Stroke Trajectory Generation for a Robotic Chinese Calligrapher

LAM, Hiu Man

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

Automation and Computer-Aided Engineering

September 2008

Thesis / Assessment Committee

Professor Yun-hui Liu (Chair)

Professor Yeung Yam (Thesis Supervisor)

Professor Kin-chuen Hui (Committee Member)

Professor James Tin-yau Kwok (External Examiner)

ABSTRACT

Robotics has been one of the significant research areas recently. Disparate to traditional machines, robots are integrations of actuators and sensors with intelligent algorithms to adapt the environmental changes. Many research groups have extended this characteristic on robots that emulate, or even surpass, human abilities. The most typical examples are manufacturing robots. This technology is also developed in different areas. In this thesis, a methodology for robotic Chinese calligraphy is proposed. It will be realized by a brush trajectory generation algorithm compatible to a 5-DOF brush manipulator. The algorithm includes three main procedures that we developed: extracting strokes from Chinese words, establishing representation of brush footprint and trajectory planning for brush movement.

Stroke extraction, which is first discussed in this thesis, is a technique to decompose a Chinese character into individual strokes. Our method first obtains the skeleton of a Chinese word to determine the features of the strokes. A fast skeletonization method based on Delaunay triangulation is introduced, which converts characters into triangular mesh. By rearranging the mesh, sample points of medial axes are obtained, which are further interpolated by Bezier curve. The advantage of Bezier interpolation is it ensures the curve smoothness and continuity in medial axes. On one hand, this approach samples the shape boundary for evenly-separated points. This infers that the method and ignores those regions that has comparatively less features. So, it is swift to extract medial axes from shapes. On the other hand, the mesh can be modified to eliminate the spurious branches and unanticipated formation of the medial axis. Therefore, stroke extraction can be achieved by segmenting the branches, which comprise the same continuous medial axes. Some rules are identified based on the structures of Chinese words. The triangular mesh of a Chinese word is modified in the suggested way. Result shows that the proposed stroke segmentation method can extract most of the strokes accurately.

For the brush footprint, it is varying with fickle brush depth and inclination. So, the footprint of a brush is analyzed with selected parameters, and the boundaries of those shapes form a discrete representation to predict such deformation. Some specific points are considered and extracted to be the features of the shapes. The variations of each feature point, caused by the change in brush orientation and pressure, are approximated using curve fitting algorithm. As a result, the points in the footprint boundary can be represented in a manner of continuous mathematical equations.

With the footprint representation and the extracted strokes of Chinese words, several criteria will be established to find a path for the brush movement. Experiment will be conducted, using the example Bada Shanren's calligraphy, to examine the feasibility of the proposed method. The result shows that this method performs well with the use of hairy brush. It can generate strokes with varying stroke thickness. Therefore, the calligraphies would be closer to the original artwork.

ABSTRACT (CHINESE)

摘要

機器人學無疑是現今重點開發的研究之一。它與傳統機器最大的分別在於機器人不單擁有驅動器及傳感器，更容許配合人工智能以達到適應實時環境變化之目的。與此同時，不少研究已著手開發人類模仿機器人；其目的是發展出有相同特性、甚至超越人類的機器人。此論文描述筆者開發的一種應用機器人實現中國書法的技術。其所得到的筆觸路徑可配合一個五軸的機器人，當中技術包括筆劃分拆、毛筆筆跡模型獲取以及書法路徑的生成。

筆劃分拆是將中國文字分解爲個別筆劃，其做法是利用文字的中心線決定筆劃的連結性。首先，利用德洛涅三角面分割演算法把文字轉化爲三角形網格，並按照特設的方法重組以使其乎合中國文字本身的結構。然後，利用變換網狀的內部邊線以達到平滑局部對稱，從而得到筆劃的橫斷線；再利用貝茲爾曲線連結橫斷線的中心點，形成筆劃的中心線。另外，由於筆劃被交叉部位分割爲多個筆劃分段，這些分段須被重組爲連續的筆劃。此重組過程是按照分段之間的差異決定，其差異包括中心線傾斜度、邊界傾斜度以及筆劃粗度。其後，在分叉部位缺少的筆劃部分會以貝茲爾曲線重構。

筆跡模型的參數是基於實驗數據所獲得。透過機器人使用毛筆把不同的筆跡畫出、並使其邊界以特定的方式簡化，再利用最小平方迴歸法將實驗得到的特徵點轉換爲特定的連貫數學模型。利用此筆跡模型可以後用作預計毛筆筆跡。而書法路徑的生成是利用局部優化的方法將毛筆筆跡放於原有筆劃之上。然後藉著調整毛筆筆跡模型的大小，令其邊界與筆劃的邊界相接觸。最後，將所得的動態毛筆筆跡串成便可構成機器人的書法路徑。此方法經已以名書法家八大山人的字畫作爲例子並進行實驗測試，演示其應用於複製中國書法之可行性。

ACKNOWLEDGEMENT

I would like to express my most sincere gratitude to my supervisor Professor Yeung Yam, who has given me the invaluable opportunity to work in the laboratory. I also thank him for his assistance, guidance, encouragement and kindly consideration, especially when I had difficulties in my research work. I am so grateful that Professor Yam has educated me a lot with not only the academic knowledge, but also the appropriate attitude to solve problems.

I sincerely thank Professor Yun-hui Liu, Professor Kin-chuen Hui and Professor James Tin-yau Kwok for serving as my graduate committee members, suggesting me many contributing comments and revising this manuscript with their patience and kindly judgment.

Many thanks go to all my close colleagues in the ICSL. I was delighted to discuss with them. Especially, I would like to thank Jeo K. W. Lo, Turkey K. W. Kwok and S. M. Wong for their experiences on the Intelligent Robotic Art System. Hearty thanks to my friends W. K. Chung, Nick T. K. Lau and T. L. Lam for their sharing my worry and happiness throughout my university life.

Finally, I would like to heartily thank my parents, Kam Por Lam and Chun Lung Ip, my aunt Yuet Mui Yip and my brothers Alan and Raymond. I am really obliged to share my worry and happiness with their love and support. Particularly, thanks to Raymond, he is always "on call" to serve as my listener.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: INTRODUCTION

1.1. OVERVIEW ON ROBOTICS

Applications of robots and robotic systems were widely developed in the last century. One major motivation of such development is to reduce humans' workload and yet attaining high productivities. Robotic systems have been treated as a universal tool in industrial production lines to enhance the quality and quantity of manufacturing. The advantages of robots include high adaptability and precision.

Up to now, robotics remains a very hot research area. Recently, researchers have designed robots with human personality, like robots with facial expressions ([1], [2]) and dancing robots ([3], [4]). A human-robot mutual communication system in [2] is an example of human emulation that includes a robot with robotic vision system. The robot has abilities to express their intention by facial expressions, gestures or speech. It would alter its intention, which determines the intention expression, based on the environmental changes recognized by the visual system. These kinds of robots may not provide apparent applications, but they would argument robots with critical abilities, like improving human-robot communication.

1.2. LITERTURE REVIEW ON ART-ROBOT

Art robot is also one of the popular topics in robotics. Several successful researches are shown in [5]-[14]. A western artwork painted by a robot is called Zanelle, which is different from other kinds of machine made arts. It is made up of brush strokes to be artist grade paintings. Many Zanelles were created with different esthetic expressions, which are indistinguishable from those made by artists. Zanelle emerged in mid 1970's. One of the representatives of art robots for

western painting is called Aaron invented by Harold Cohen ([5] and [6]). Aaron is a pioneer in robot art with artificial intelligence. Many of its artworks have been exhibited in museums and give great impression to the public.

Figure 2. A photo of Drawing Machine

Another robot, called Drawing Machine [7], is developed by Fernando Orellana. A photograph of the robot is shown in Figure 2. This robot is capable to create abstract drawings. A microphone is installed in the robot to receive environmental sound. The sound would then be interpreted to commands to depict complicated drawings. Randomness is also added to increasing the variation of the artworks. With the "listening" ability, the viewers may cooperate to create artworks. When they talk, sing or make some noise near the microphone, the robot would receive the sound signals. Drawings are created based on these "human sounds".

Figure 1. The art robot Zanelle

Zanelle (Figure 1) is a painting art robot for creating chromatic drawings from photographs [8]. Similar to ordinary art robot, it is a 3-DOF robot with a manipulator to control the brush motion. The robot can drawing with different colors. Color ink containers are put at the side of the robot. The brush can automatically refill ink or change color. One of the features of its artworks is that the drawings are composed of colored straight lines as shown in the photo. However, the output would depict the contour of the original image.

1.3. ROBOT ARTIST FOR CHINESE CALLIGRAPHY AND PAINTINGS

An art robot, called Intelligent Robotic Art System (IRAS), was developed in the Intelligent Control System Laboratory, CUHK, by Prof. Yeung Yam and Joe K.W. Lo [13]. It was an artistic robot with 3-DOFs (x, y and z) to perform professional line drawings using non-hairy brushes. One of the artworks, which were done by the author in 2006, made by IRAS is shown in Figure 3. It is a gray-scale painting that replicated from a chromatic Chinese painting. Recently, the robot is modified to be a 5-DOF robot, by adding 2 additional rotary joints (control brush inclination), to improve its flexibility. Hairy brush is used to draw brush strokes, so IRAS has potential to create artworks, like calligraphies and paintings. The shapes of the brush footprints are modeled using experimental data. It would be changed by variations in position and inclination of the brush.

(a) (b) (c)

Figure 3. Grayscale image drawn by IRAS. (a) The original painting, the image (b) converted to be gray scale, and (c) image of the grayscale painting by IRAS.

1.4. MOTIVATION AND RESEARCH OBJECTIVE

Robotic replication of Chinese calligraphy is a great challenge in robot emulation of human abilities. Many robots have been developed to mimic human-made artworks. However, unequal to these artworks, Chinese calligraphy is controlled by complicated hand motions, which controls the thickness and continuity in strokes. So, augmentation with additional DOFs is essential to improve the brush manipulation.

One of the difficulties in this project is how to segment strokes automatically from a Chinese word. Segmentation of strokes in calligraphy would be a complicated problem, since the thick strokes would change the shapes of words. So, the structure of calligraphy should be studied to generalize the common features. Contour analysis of a Chinese word would be a feasible method. Medial axes can be generated as additional information to descript a shape. Hence, it should be utilized to determine the continuity of strokes.

On the other hand, mathematical representation of brush footprints should be developed to predict its shape variation when the hairy brush deforms. Such representation should be in form of continuous functions, since the shape of brush footprint changes by sequence of motion executed by the manipulator. Moreover, it should be compatible to fast searching method to precipitate the trajectory generation process.

In this project, the major objective is to study and replicate Chinese calligraphy using engineering technology. Due to the repeatability of the robot, the features and essence of calligraphies written by artists can be permanently preserved. For this reason, many invaluable calligraphies, which are important cultural heritages, can be regenerated using this approach.

1.5. THESIS OUTLINE

In this thesis, a methodology for calligraphic stroke trajectory planning is introduced. Chapter 1 introduces some art robot developed by different research groups. Then, in Chapter 2, the configuration of Intelligent Robotic Art System developed by the intelligent control system laboratory is introduced. The thesis continues to talk about the artworks done by the art robot to show its performances feasibility for duplicating existing artworks and even creating new art styles. As the robot is capable of writing Chinese calligraphy recently, some new mechanisms and software are added to advance the function on the basis of the developed robot. The new functions includes: two additional DOFs for rotational motion of the manipulator, infra-red sensing system for manipulator homing.

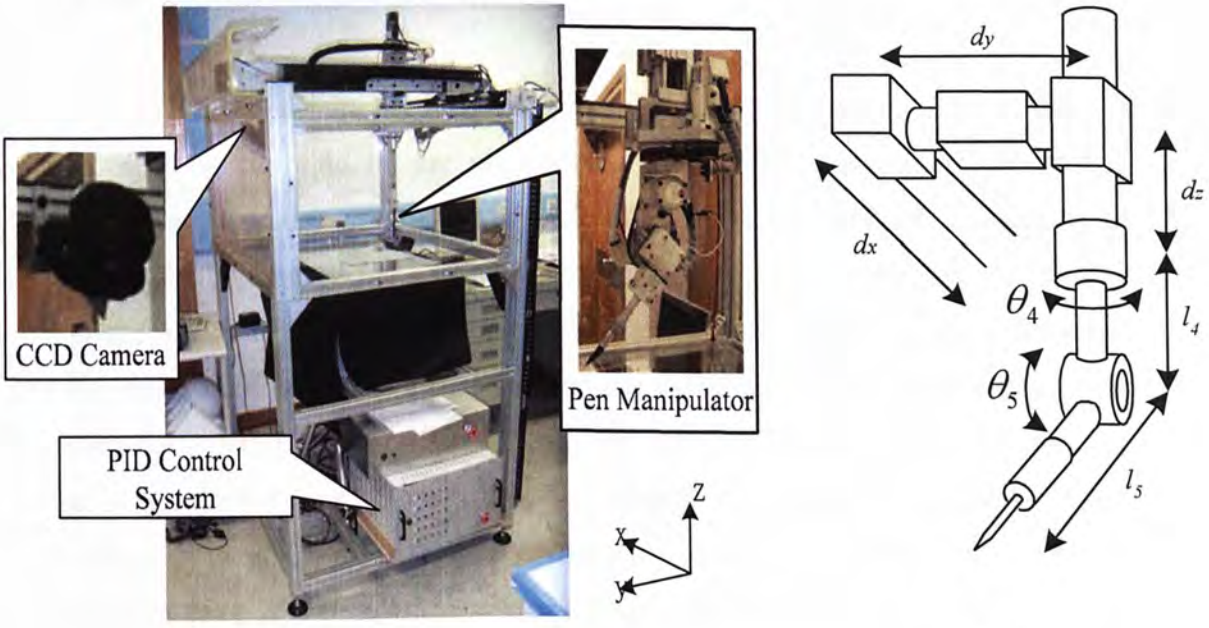
As the original system uses brushes, a non-hairy brush, to delineate line drawings and chromatic images, new procedures were developed to adapt high deformable hairy brushes. Series of algorithms are proposed to achieve robotic Chinese Calligraphy. Stroke segmentation method is first discussed in Chapter 3 and Chapter 4. The method is originally for medial extraction. It is further modified to segment Chinese words into strokes based on several rules summarized from the general features of Chinese calligraphy.

Since a hairy brush is essential for calligraphy, the brush footprint model acquisition is also discussed, which is approximation to shapes of brush footprints. The method for obtaining the footprint information and generate the model is discussed in Chapter 5. Then, Chapter 6 proposes an approach to search trajectories, which would be executed by IRAS, using the brush footprint model and the segmented strokes.

Chapter 2: INTELLIGENT ROBOTIC ART SYSTEM

2.1. PREVIOUS CONFIGURATION

Manual drawing/writing can be viewed as the pen/brush movement, manipulated by a human hand, along a trajectory in certain pattern. The line and shade in a picture are produced by sliding the pen tip on the target surface. A robotic system must be able to generate identical pen movements of a painting artist/calligrapher for it to eventually perform professional artistic tasks. Our research group has recently developed an Intelligent Robotic Art System (IRAS) as shown in Figure 4.



(a) (b)

Figure 4. (a) IRAS and (b) its schematic.

A clip is located at the terminal of manipulator and so that different types of drawing tools, such as pencil, paint brush, or even Chinese writing brush, can be mounted onto the manipulator for different drawing styles. When the robot writes a word, a paper can simply be fixed on the stage. Upon setting the origin

coordinates based on the position of the stage, the predefined image/word can be drawn on the horizontal surface automatically.

2.1.1. 3 DOF Manipulator

The platform contains a robotic manipulator with 3 translational joints in x, y and z directions. The corresponding workspace size is 700 mm (x) × 800 mm (y) × 150 mm (z). The resolution of movement in each axis is 0.1 mm, which can provide precise and high quality picture drawing. To enhance the output strokes smoothness, a PID controller is used in the system positioning to minimize the superfluous oscillation during operation.

2.1.2. Digital Image Input System

Two CCD cameras are used to capture images on the glass plane. One of them is mounted at the side of the platform to capture the drawing procedure and outcomes. It is also used as a vision feedback device to improve the quality of mechanical outputs. After a word is written, the camera would capture the resulted image, which is then transformed to front view. As the position and orientation of the camera are not at the top of the drawing surface and there is distortion in the camera caused by the refraction of lens, the captured image may not directly used as a resulted image. Homography is applied to correct these deformations. However, parameters for the transformation would not be easily obtained due to error existing in the experiment, e.g. reference points input, and experiments are iteratively conducted to reduce the uncertainties. Afterwards, the front-view transformed image is then compared with the original image for analysis. Based on GA-based iterative improvement algorithm, it searches the brush pressure, which generated the line thickness that is closest to the source image. Finally, the mechanical outputs would be the same as the original source images.

Besides, another camera installed at bottom part of the robot is used to capture the footprints of hairy brushes. In Chinese calligraphy and painting, stroke thickness affects the aesthetic feeling of the entire artwork. Therefore, this camera provides a preliminary way to achieve robotic art using hairy brushes.

2.2. HARDWARE MODIFICATION

2.2.1. Additional Degree of Freedoms

The former configuration design provides a stable and efficient performance for artworks, which only require motion in x, y and z direction, e.g. line drawing. However, for some other artworks, like Chinese calligraphy, require using brush to intensify the aesthetic perception to spectators, more DOFs are required to control the brush motion and the desired shape of brush footprints. However, due to the limitation in flexibility, the 3-DOF robot may not perform well in particular motions. Recently, to illuminate such problem, it was modified to be a 5-DOF robot with the original translational joins and three additional rotary joints, which is illustrated in Figure 4(b). The additional joints are z-rotation, pitch and in direction parallel to the brush. The function of the first two joints is enhancing the manipulation of the pen manipulator.

However, the motion control interface for the previous configuration does not adapt to systems with more than 4-DOFs. Therefore, Galil PCI bus controller DMC1886 is replaced, which is shown in Figure 5. It is an interface card for fast motion control. It provides 8 channels for 8 individual servomotors. It also supports PID control for vibration reduction. Moreover, this motion controller provides linear motion manipulation. Hence, all the axes of a system are synchronized, which means the axes would range to the desired position and stop at the same time. This function is essential for the new 5-DOF design, since it improves the system in the accuracy of motion. Especially the motions with large direction changes, they can also be ensured to be smooth in high speed motions.

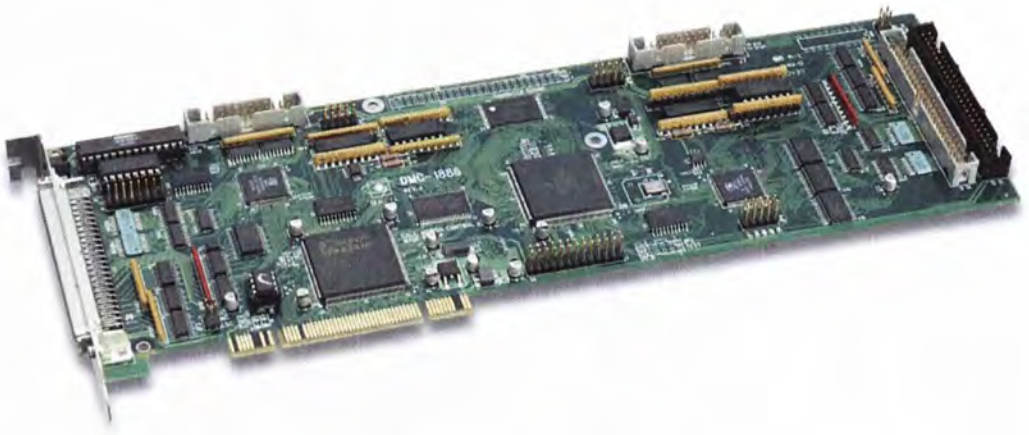


Figure 5. Galil PCI Bush Motion Controller DMC1886

2.2.2. Infra-red Sensing System for Manipulator Positioning

In the previous robot configuration, it does not have sensors to fix manipulator origin, which means it does not have constant position calibration. So, the origin is set to be the manipulator position when the robot is activated or varied by users. In this case, the robot would have higher flexibility. However, it cannot repeat its operations after it has been switched off, it would hence critically reduce the robot repeatability. Therefore, positioning is significant to art robots, as artworks would be finished in separate periods.

In the new art robot design, as two rotational axes is added, sensors are added to totally 5 DOFs (x, y, z, z-rotation and pitch). Figure 6 shows the circuit diagram for the sensors. Infra-red (IR) sensor H21A2 is used, which comprises an infra-red LED and a photo-sensing resistor. So, IR is directly emitted to the IR receiving part and gives a high signal. A low signal is detected, if the communication between the IR source receiving part is blocked.

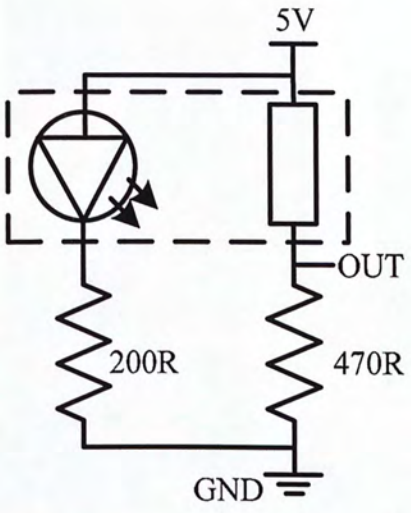


Figure 6. Circuit diagram for the position sensors

The sensors are mounted on the platform of IRAS at the position indicated in Figure 7. When the manipulator of IRAS moves to particular positions, a thin slice, which connected to the manipulator, blocks the emitted IR. There is no signal transmitted in the sensors. The system will recognize that the manipulator is at certain position and set it to be the system origin. So, the system can still constant coordinate frame after it is off.

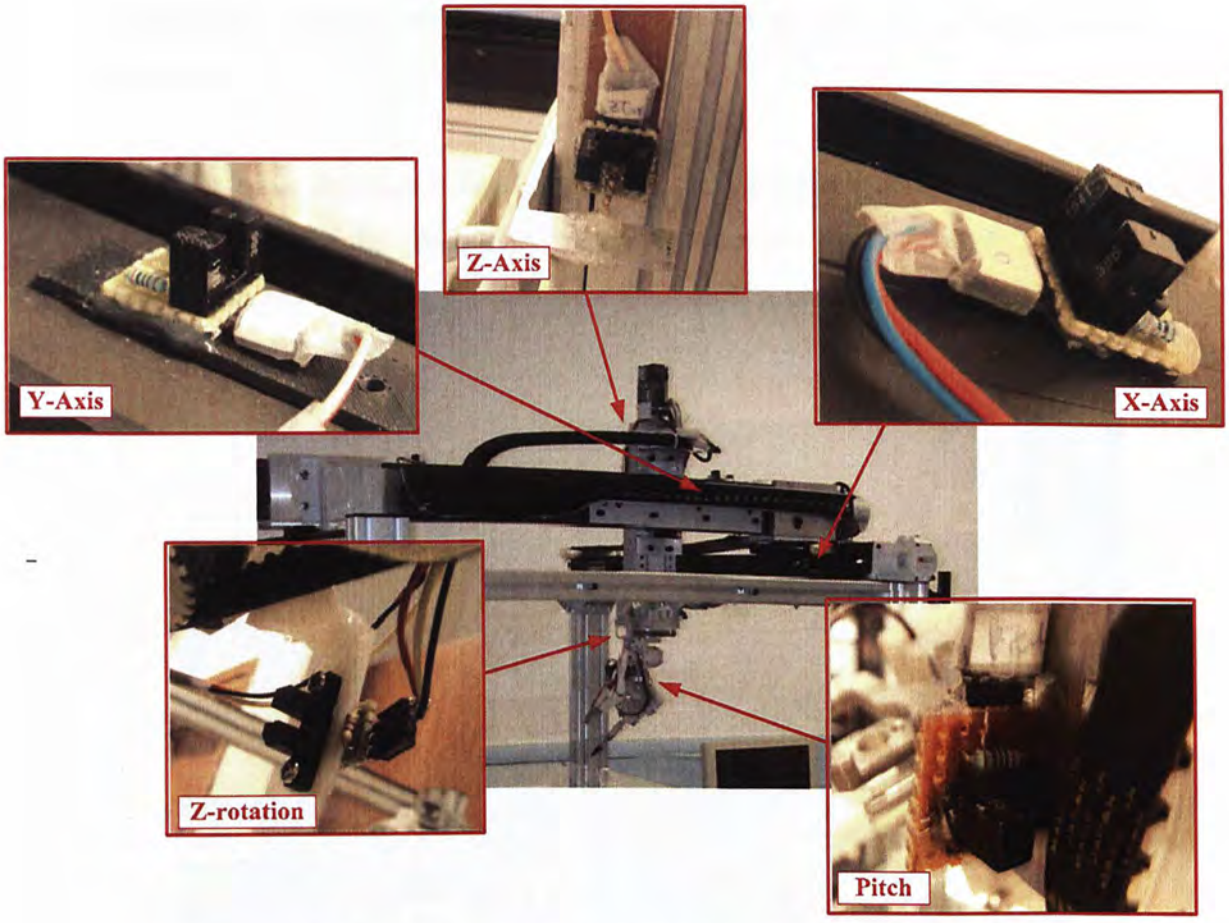


Figure 7. Infra-red sensors in IRAS for positioning of the manipulator

2.2.3. Axial-rotary Brush

Apart from the rotational axes, an axial rotator brush is also installed on the manipulator. It is designed to rotate the brush about axis of its brush shaft (longitudinal axis). This rotational motion is essential for using hairy brush, since calligraphists rotate a brush to ink and retain the hair to be straight. Figure 8(a) shows the design of the brush holder. Screws are inserted to the screws holes SO_1, SO_2 and SO_3 to fix the position of the brush, which is put into the brush hole. The screw holes are arranged in two layers to enable fine adjustment of the brush orientation. Figure 8(b) is the part to embed a servomotor BMS-380 in the chamber. The brush holder is joined to the servomotor as indicated. When the

servomotor operates, the brush holder is rotated and the brush-shaft rotation is actuated.

For this joint, although it cannot improve the workspace of the robot, it enables different sides of the brush hair contact with the horizontal surface. Therefore, it can be utilized in ink refill and brush refinement. A photo of the axial rotary brush is shown in Figure 9.

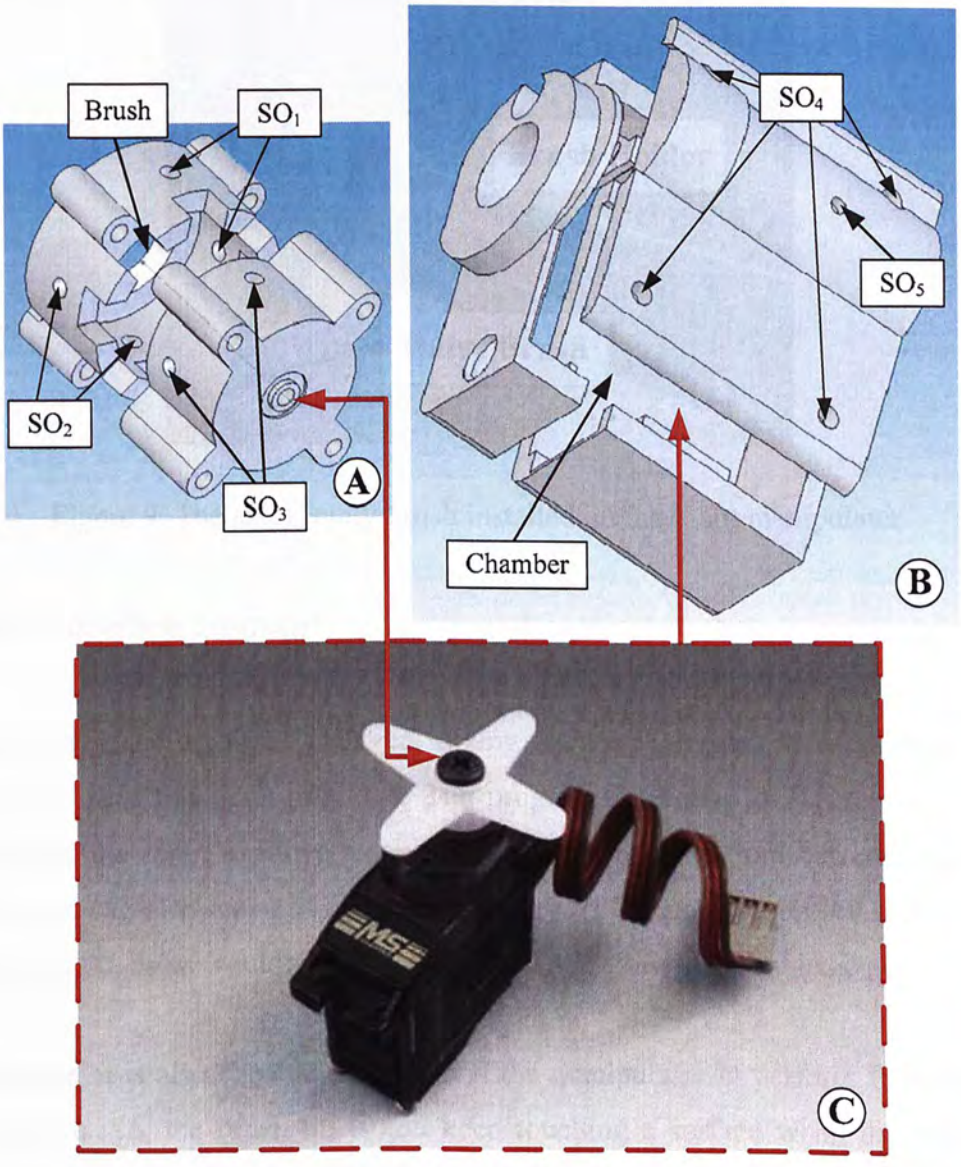


Figure 8. Axial rotary brush holder: (a) brush holder, (b) motor holder and (c) the mini-servo for rotational motion

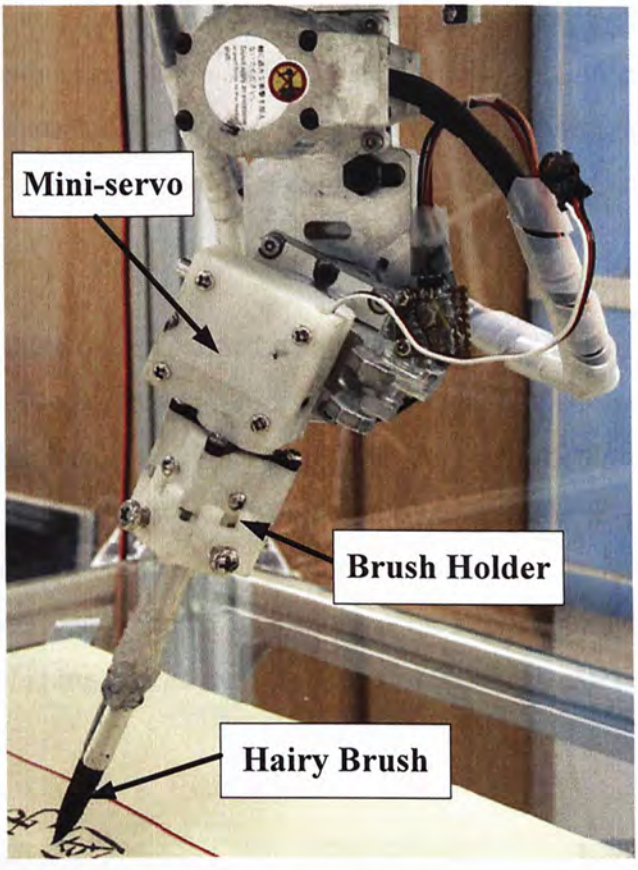


Figure 9. The axial rotary brush installed on the robot manipulator.

2.2.4. Interface program

As the interface card was changed, a new interface program was developed to adapt the new functions provided. The program is shown in Figure 10. It can command the robot to move to position as input or shift from current position. The robot can also move in planned trajectory. It is displayed on left side of the interface. So, users would view the words/image before the robot executes.

Moreover, it is also capable to command the manipulator to perform fixed brush tip motion. So, the brush tip would keep touching a surface when the brush is moving. This function is important in writing Chinese calligraphy. It is because when a brush is pressed on a surface, the hairy part of the brush deforms. A brush

footprint is generated that the shape of which is determined by the incline angle and the brush depth insert to the surface. The fixed tip motion enables the brush manipulator to create footprint using the same reference point (brush tip). Then, the actual position of the footprint is controlled by the manipulation in x and y directions.

IRAS can also follow the predefined path to ink the brush manipulator. Figure 11 shows the steps of the inking process. The brush is first moved to the ink container, such that it can absorb the ink inside. The servomotor in the axial-rotary brush holder is activated, which would improve the ink absorption. Then, it reaches to the ink absorber (Figure 11(b)) to remove the excessive ink. Finally, the brush hair is straightened, when it is moving on a plan surface (Figure 11(c)). With this brush shaping procedures, the shape of the brush can be controlled after drawing strokes.

Figure 10. The interface program for IRAS

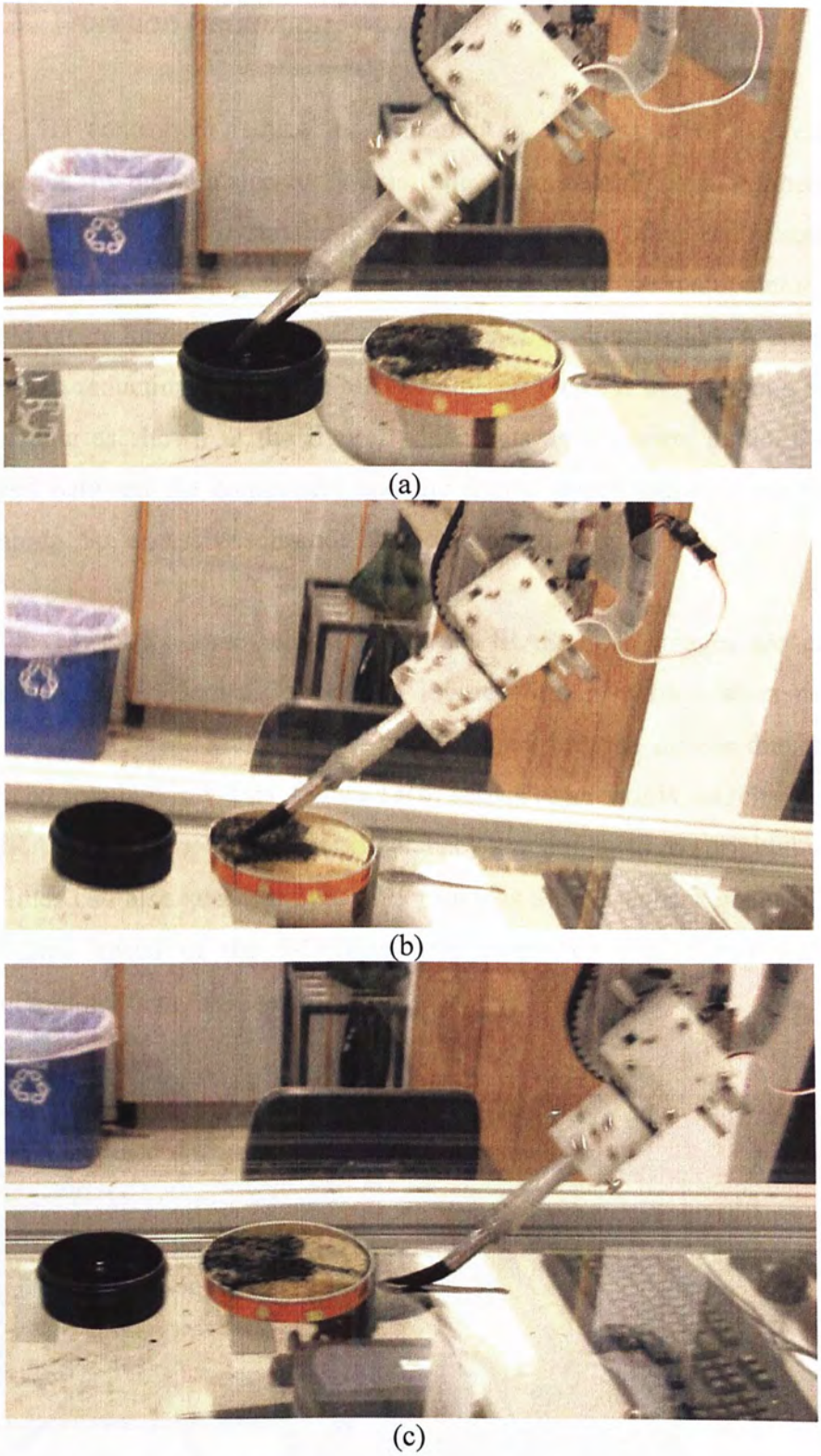


Figure 11. Brush inking procedures: (a) ink the brush, (b) absorb excessive ink and (c) straighten the brush hair

2.2.5. Vibration Reduction

The PID controllers reduce the vibration in the motion of the manipulator. However, in the actuator for y-axis, one of the terminals is connected to the actuator of x-axis. It is stabilized by PID control. The other end is suspended on the frame of the configuration. If the acceleration of the manipulator is high, it would create large vibration that cannot be stop in short period. A component for vibration reduction is added to alleviate the problem. It moves along the track in the frame as shown in the figure, which acts as a motion guide. Sponges are placed between the component and the frame, which increase the friction and damping. So, excessive vibration can be absorbed.

Figure 13 shows some words written by IRAS. Both images are executed in maximum speed 5cm/s. The one in Figure 13(a) is written when no vibration reduction is made. Many of the straight lines become curves due to the non-damped terminal in y axis. Figure 13(b) shows other words, which is also written by IRAS with damping part installed. They are similar to the original version and the lines can also keep straight. This provides a great enhancement in maximum execution speed of the 5-DOF system from 1cm/s to 5cm/s with constant acceleration $10\text{cm}/\text{s}^2$.

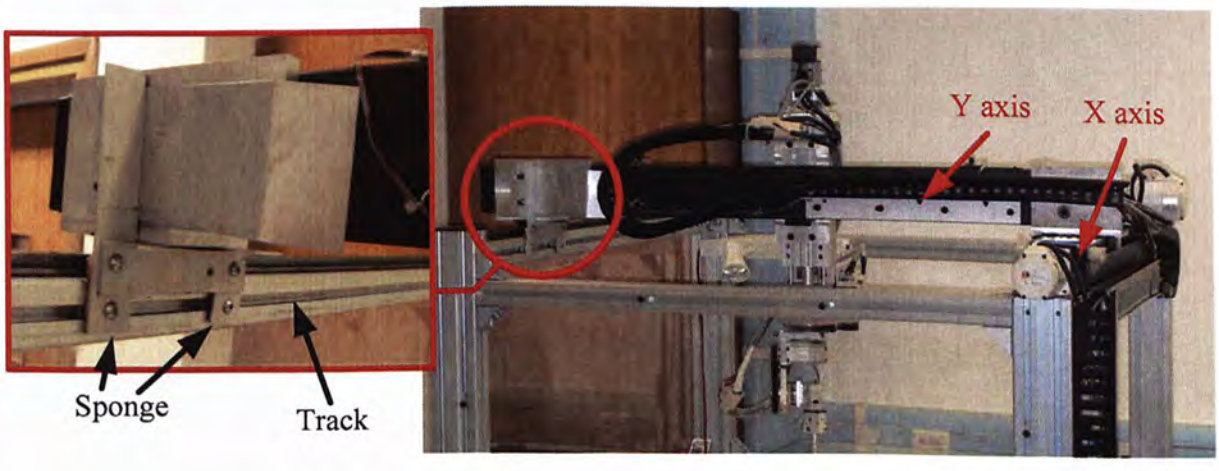
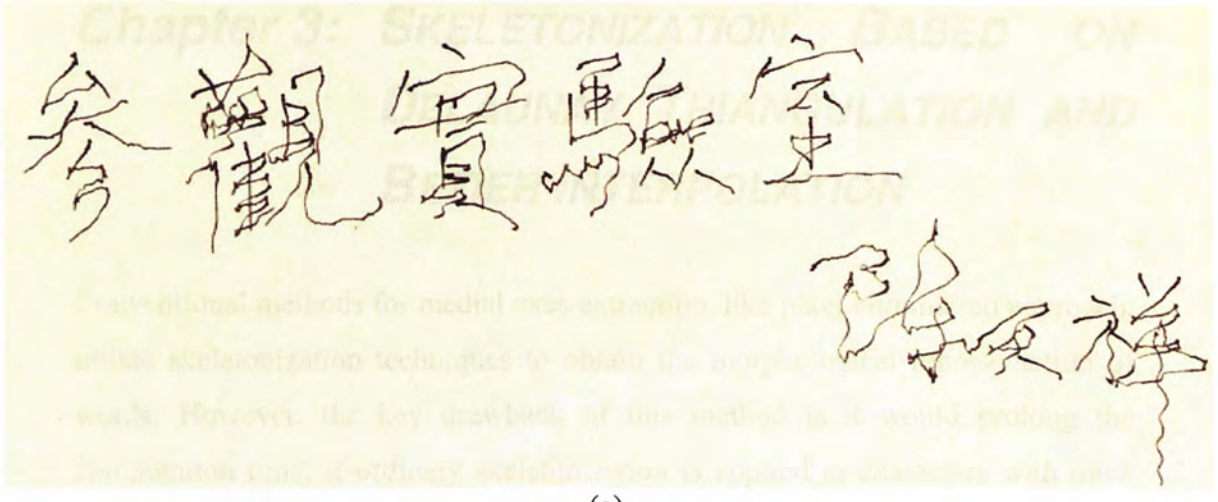
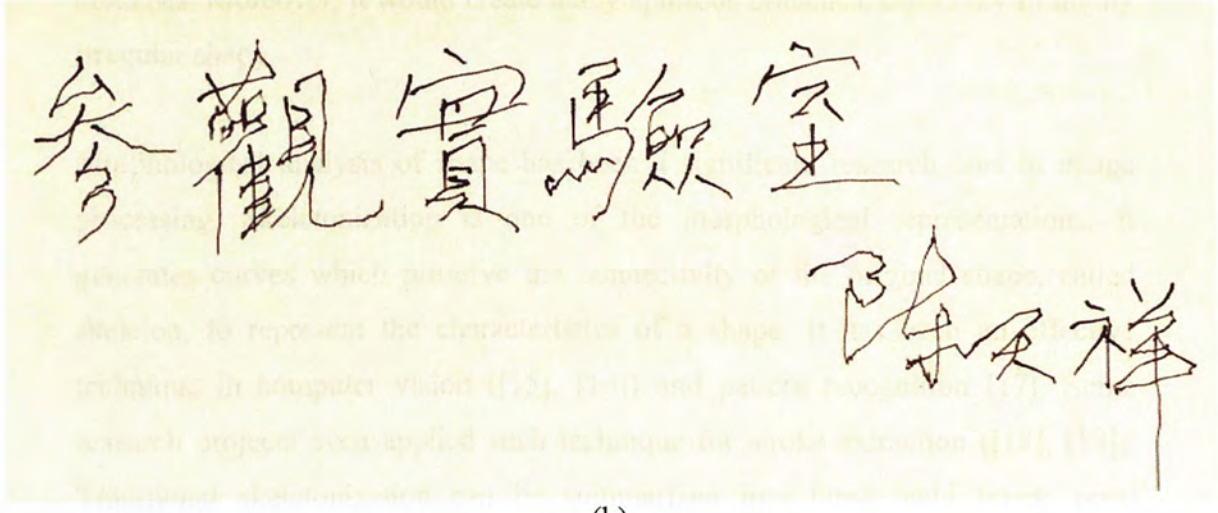


Figure 12. Vibration reduction part installed in the y-axis.



(a)



(b)

Figure 13. Chinese words written by IRAS with predefined trajectory: (a) without damping part and (b) with damping part.

Chapter 3: SKELETONIZATION BASED ON DELAUNAY TRIANGULATION AND BEZIER INTERPOLATION

Conventional methods for medial axes extraction, like pixel elimination approach, utilize skeletonization techniques to obtain the morphological representation of words. However, the key drawback of this method is it would prolong the computation time, if ordinary skeletonization is applied to characters with thick branches. Moreover, it would create many spurious branches, especially in highly irregular shape.

Morphological analysis of shape has been a significant research area in image processing. Skeletonization is one of the morphological representations. It generates curves which preserve the connectivity of the original shape, called skeleton, to represent the characteristics of a shape. It has been an effective technique in computer vision ([15], [16]) and pattern recognition [17]. Some research projects even applied such technique for stroke extraction ([18], [19]). Traditional skeletonization can be summarized into three main types: pixel elimination [20], model based skeleton matching ([20], [21]) and contour analysis ([22]-[26]). Pixel elimination and model based approaches are effective to skeletonize low resolution images, while contour analysis is often used to extract skeleton from large images.

The pixel elimination approach iteratively erodes pixels in the shape boundaries by the line thinning algorithm. Some pixels are recognized as the shape characteristics and are retained to form the skeleton. The main advantage of the method is simple to implement. It can generally extract a smooth skeleton. However, a high resolution image usually contains shape regions with a large number of pixels. It may require more iterations to erode the pixels, and so the computational time is too long to extract a skeleton. Also, ragged edge would

cause spurious branches. This leads to inconsistency and inaccurate detection in pattern recognition.

For model based skeleton matching, a shape is segmented into partitions. A group of patterns is utilized as morphological image model. Each pattern has a predefined skeleton. Comparing the shape with the image model, if a shape partition is approximate to one of the predefined patterns, the corresponding skeleton is mapped to the partition. The skeleton fragments are then connected to reconstruct the recognized shape. The main drawback of this method is the limitation on the flexibility of predefined image model. It is not applicable to images with inconsistent scales. Moreover, a high resolution image has higher complexity in terms of pixel distribution. It requires extensive image patterns for comparison and considerable number of predefined skeletons for the shape reconstruction.

The skeletonization of contour analysis depends on the medial (or symmetric) axis transformation. First, the boundary is extracted and the pattern branches are segmented. The skeleton is then obtained by searching the local symmetry of edges. The locus of symmetry points is extracted to be the medial axis. These axes are connected in nodes to form the skeleton. The advantage of this approach is that it can generate an accurate skeleton, which contains no excessive branch. However, it requires extensive sample points for the skeleton estimation. At the same time, it increases the complexity of calculation and the processing time would be relatively slow.

In this chapter, a skeletonization method is proposed, which is an efficient method applicable to high resolution planar shapes. The algorithm presented in this paper is a medial axis transformation approach, which assures the skeletal accuracy. Instead of linear interpolation, Bezier curve is used to connect the inconsecutive sample points. It can approximate the gradient change of the symmetry point locus. So, the proposed method is applicable to skeleton gradient dependent applications.

3.1. BACKGROUND THEORY

3.1.1. SMOOTHED LOCAL SYMMETRY

Skeletonization of contour analysis applies the concept of local symmetry theory ([27]-[31]) which is a significant theory in image processing, computer vision and computer graphics for shape representation. The conceptual figure is shown in Figure 14(a). The theory suggests that local symmetry is a subset of a shape that two points A and B (red dots) in the contour are symmetrical about normal N from the mid-point P_s (blue dot) of line AB. In other words, when we map circles into a shape, each of them intercepts the shape contour at two points. Therefore, the tangents of the circle at points A and B form the same angles to the line AB. The center of the circles can be connected to form a locus, as highlighted with green in the figure. The center of the circle is the intersection point of normal from points A and B (N_A and N_B). For a ribbon-like pattern, this locus splits a shape into two halves with corresponding symmetrical features. And, since the locus interpolates the mid-points of local symmetry point parts, it is used in medial axis transform (MAT), also called symmetric axis transform, to find the medial axis of a shape.

Smoothed local symmetry (SLS) is a type of the local symmetries, as shown in Figure 14(b). It is the relation between two edges, which has similar criteria to local symmetry:

- a supposititious circle intercepts both edges
- the edge gradient and the circle gradient are the same at the corresponding point of interceptions

One thing different from local symmetry is that SLS extracts symmetric point P_s, which is the center of the interception points A and B in each edge. The locus of symmetry points is extracted to be the shape skeleton. Though both methods interpolate different symmetric points, they would give almost the same result in

most cases. The advantage of SLS is that it is not required to find the circle touching the point pairs, which effectively reduce computational time.

However, the point pairs of local symmetry may not be easily obtained. It is a very time-consuming process to search along the shape boundary. To increase the efficiency of MAT, some research group has proposed to use the edges of triangles to approximate the cross-section line AB in Figure 14(b) [24]. This triangular approximation is shown in Figure 14(c). Cc is a circum-circle, which intercepts the triangle at three vertices. Consider that if the triangle edge S is infinitely small, it shrinks as a point. This means the internal edges between point P and edge S become SLS. Cc also has tangent with same gradient to the shape at the point pair (P and S). In general, if S is short, we also consider the edges of the triangle satisfy SLS, as its length cannot be infinitely small. So, a line M connects the mid-points of the internal edges to form the medial axis.

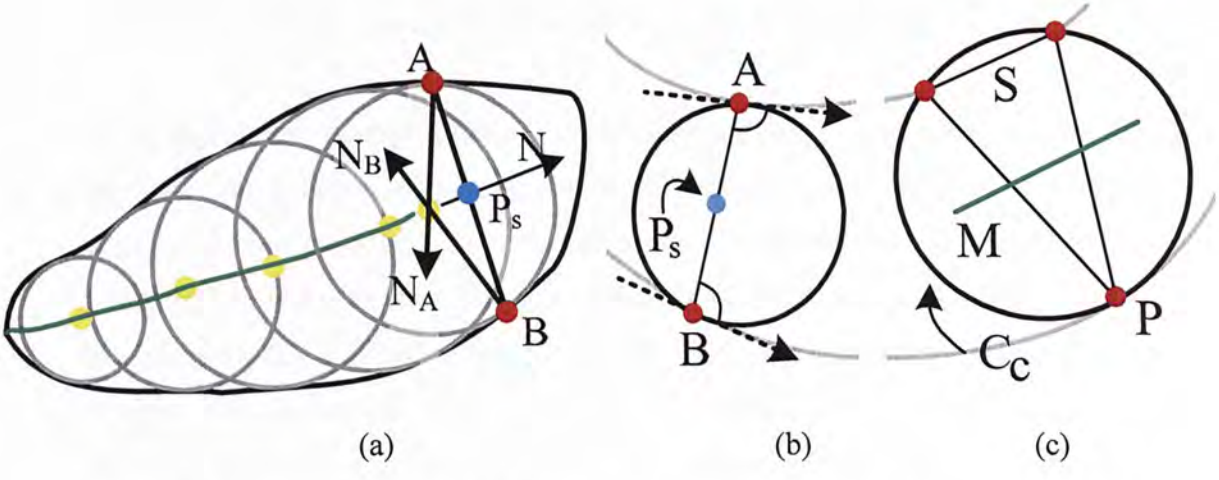


Figure 14. Illustration of (a) local symmetry, (b) smoothed local symmetry and (b) Delaunay triangulation

3.1.2. DELAUNAY TRIANGULATION

Delaunay triangulation (DT) is a method to convert a set of Cartesian coordinates into triangular mesh. It maps circumcircles to vertexes, such that each circle

intersects with the boundary at three vertexes. The vertexes connect mutually to form a triangle, providing that there is no other vertex inside. An algorithm called divide and conquer is capable to convert a set of points into triangular mesh. It operates by recursively split the points into two groups according to their y-coordinates. Triangles are inserted to the region between the convex hulls of the groups. When the points separated to be triangles, they are merged to form a triangular mesh that bounded by the convex hull of the points.

Some research projects extended this method using constrained Delaunay triangulation (CDT) [19] (illustrated in Figure 14(c)). However, this approach requires high density of sample points along the boundary. The computational time for DT is O(nlog(n)). In a high resolution image, DT may segment a sampled pattern into a considerable number of triangles. If the number of sample points is too large, the time for skeletonization would be too long for online applications; if the number of sample is too small, the information may not be sufficient to estimate the skeleton and cause the skeletal distortion.

After appropriate triangle filtering, the remaining triangles are enveloped by the boundary and fill compactly in the pattern. The triangles are composed of external edges and internal edges. An external edge is an isolated edge located at the boundary while an internal edge is a common edge of two triangles. The interior triangles can be classified to be end triangle (ET), normal triangle (NT) and joint triangle (JT) [24] as illustrated in Figure 15. An ET contains only one internal edge. An isolated corner is not connected to this internal edge. It is a terminal of the skeleton. An NT has two internal edges. It is a transitional triangle which provides a connection path for the skeleton. A JT has three internal edges. It combines with the adjoining JTs to become an intersection region. In the intersection region, there should be a node connecting the skeleton from the adjoining branches.

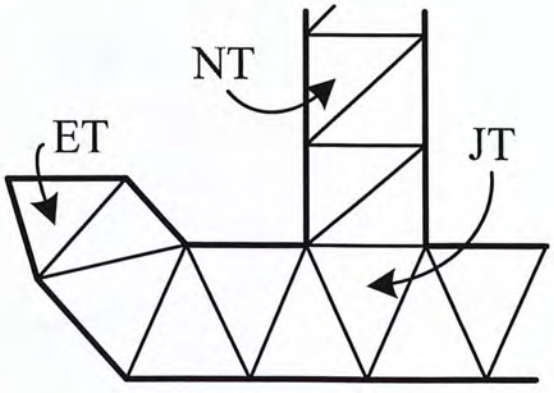


Figure 15. An interior triangular mesh.

3.1.3. Bezier Curve

Bezier curve is a technique to produce curves, which are smooth and can be scaled. It is widely used to model smooth curves, especially in computer graphics. The curve starts at the first control point and ends at the last control point. The gradient of it at the starting point and ending point can be controlled the control points. However, no matter how the control points are modified, the curve is also enveloped by the convex hull of the control points.

The idea of Bezier curve is illustrated in Figure 16. The blue circles indicate the control points for the Bezier curve, the gray triangle is the convex hull and the red curve is the generated Bezier curve. Advantage of quadratic Bezier curve for skeletal interpolation is that the initial and end gradients of the curve follows the direction C_2-C_1 and C_3-C_1 respectively. The general form of Bezier curve is

$$\sum_{i=0}^{n} P_i \binom{n}{i} t^{n-i}(1-t)^i, \qquad for\ t \in [0,1] \qquad (2)$$

where Pi are the control points, t controls the position of point along the curve and n is the order of the Bezier curve.

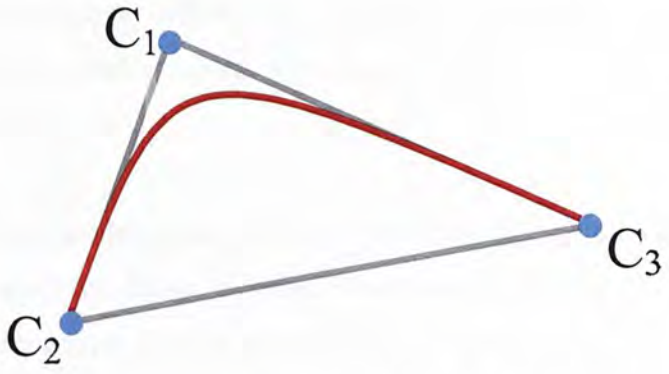


Figure 16. Illustration of quadratic Bezier curve

3.2. ALGORITHM

This skeletonization is based on approximation of pattern. The algorithm has several procedures, including edge sampling, triangulation and mesh modification, triangle filtering and replacement, internal edge refinement and skeletal interpolation.

3.2.1. Edge Sampling

The shape boundary is extracted and discretized by uniform sampling. If the sampling interval is too large, the features of a shape may be mistakenly neglected. These features are the boundary turning nodes and should be extracted independently. They are located at sharp corners, which have the local maximum of gradient change. The gradient change of boundary can be approximated as:

$$\Delta B(t) = (B(t+s) - 2B(t) + B(t-s))/s^2, \quad \textit{for } t=1, 2, 3...n \qquad (1)$$

where B(t) is the function of vertexes in the pattern boundary and s is scalar parameter representing the sampling interval. However, there exist many teeth in a lumpy shape, causing ambiguous indication of turning nodes. To eliminate such effect, those of small gradient change should be excluded. The remaining turning nodes are classified as concave or convex corners. A convex corner is the pinnacle of a salient. It is also a branch terminal of a skeleton. A concave corner is the tip

of a re-entrant boundary, which is the vertex of an intersection region. The boundary is split by turning nodes. The boundary between two turning nodes is called edge segment.

Convex corners are the terminals of branches. JTs should be created between the branches for connection. Along the edge, if a sample point is not found in an edge segment, JT cannot form to be a joint of the skeletal branches. This means any cascaded turning node must be prevented. A mid point in the edge segment is inserted to be an additional sample point. Besides, there would be some closed pattern, which has no sharp corner. Figure 17(b) shows two examples, which are circle and a slick shape. These shapes have no apparent connectivity. The pattern center is extracted to be a point skeleton.

Then, the skeletal axis is obtained using the sampled boundary. However, the locus of symmetry points cannot be extracted directly. The sampled points may not find its counterpart to accord SLS. They are triangulated to segment the shape. The fragments are further processed to improve the accuracy of medial axis extraction.

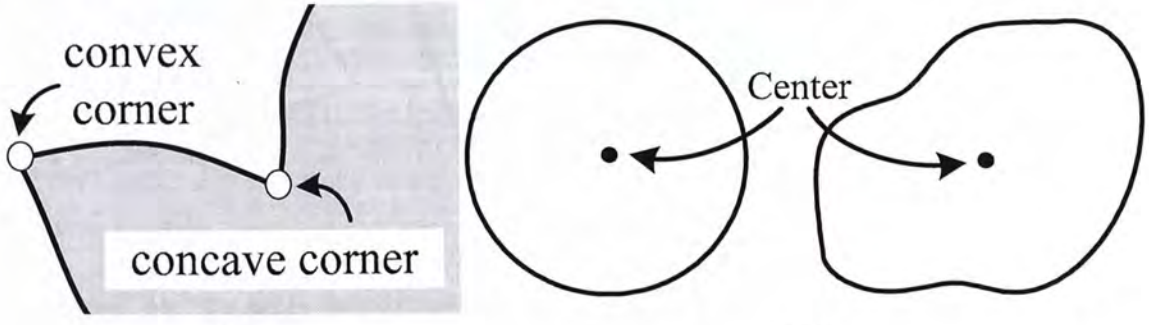
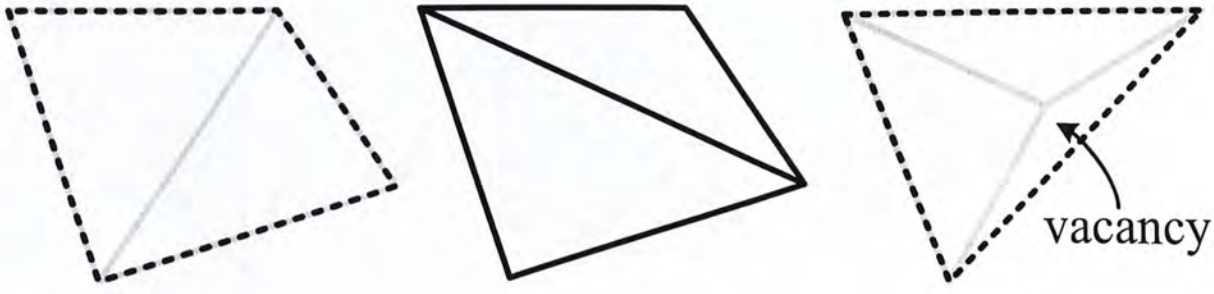


Figure 17. (a) Two types of turning nodes: convex and concave corners and (b) examples of shape with no turning node

3.2.2. Triangle Modification

After triangulating the sample points, the triangular mesh may not be created properly as expected. Some triangles ambiguously separate the branches and this critically reduces the quality of a skeleton. Especially in CDT, many triangles would form in the same edge segment. So, the triangles should be reorganized partially.

Triangle modification, also called flipping algorithm, is applied to two adjoining triangles for internal edge alternation and vertexes interchange. For instance, the triangles must have a convex hull (dot line) formed by all the four vertexes in Figure 18(a_1). The triangles are first merged into a quadrilateral and are re-sliced into two triangles along the isolated corners. A modified triangle pair is shown in Figure 18(a_2). In case the convex hull is a triangle and there is a vacancy in it (Figure 18(b)), the interior triangles are not able to merge and re-slice into two separate triangles.



(a_1) (a_2) (b)

Figure 18. Triangle modification: (a_1) convex hull of the triangle pair, (a_2) modified triangles in Figure 18a_1 and (b) triangle pair with a vacancy

Three typical cases of triangle modification are shown in Figure 19. The white circles indicate the concave and convex corners and the black circles are the regular sample points. The first case occurs in the intersection region (Figure 19(a_1)). The gray triangles enlarge the intersection region and the skeleton

branches may join in the position. It should be modified to be that in Figure 19(a_2). In Figure 19(b_1), more than one triangle comprises the convex corner. Two ETs appear which leads to superfluous skeleton branches. The triangle modification should reduce one of the triangles connecting to the convex corner. If there are n triangles connected to the convex center, the process should be performed n-1 times so that the convex center becomes the isolated vertex in the ET (Figure 19(b_2)). In some cases, an excessive JT appears due to a spurious branch (Figure 19(c_1)). The JT is modified tautologically with its neighbor triangles in the spurious branch. The resultant should have no JT as shown in Figure 19(c_2)

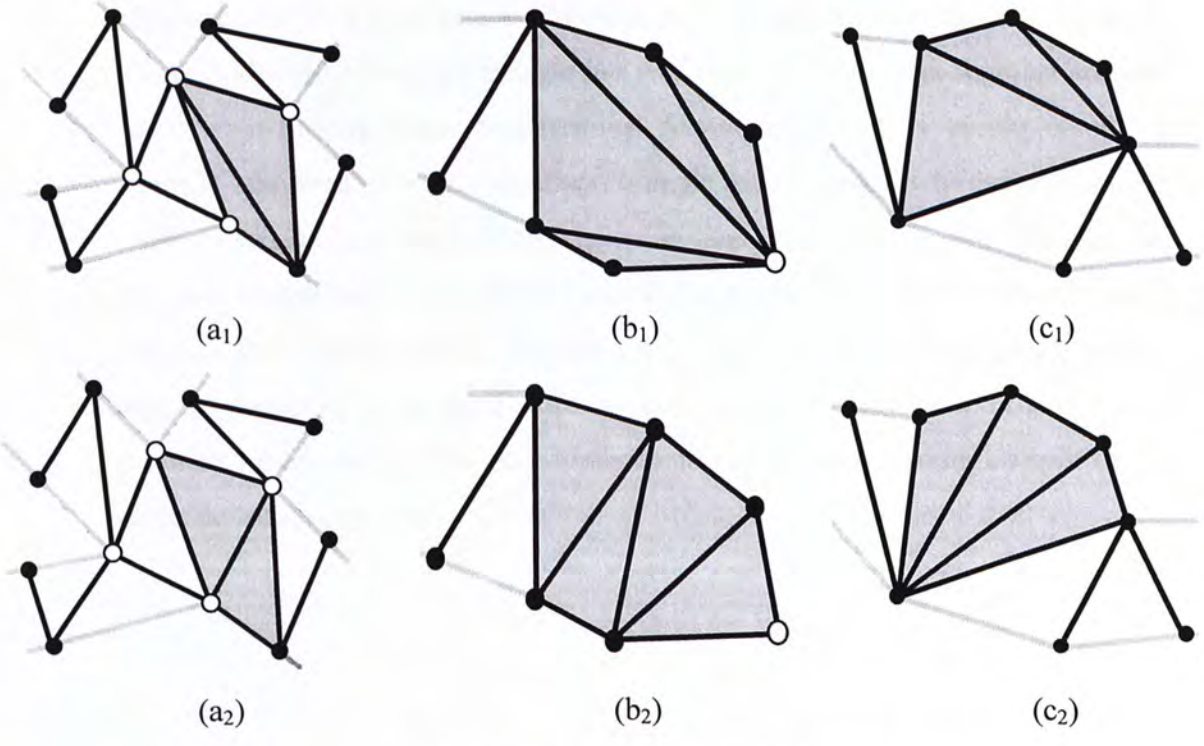


Figure 19. Three types of triangles modification: (a) intersection region reduction, (b) elimination of inappropriate triangles in convex corner and (c) removal of spurious branch

3.2.3. Triangle Filtering and Replacement

In the triangulation process, triangles would be formed outside the pattern boundary. Most of these exterior triangles can be removed if their centroids are excluded from the pattern. The rest of them are failed to be eliminated because of the irregular boundary. Exterior triangles are redundant triangles laid on the boundary and have two internal edges. They can be removed following specific rules. The possible configurations of the exterior triangles are shown in Figure 20.

Figure 20(a) shows an exterior triangle (gray) that has three vertexes in the same edge segment. This may induce two excessive JTs and cause spurious branches. In Figure 20(b), the redundant triangle has two vertexes in an edge segment and one vertex at a turning node. The terminal feature of a convex corner would be wrongly removed. For a redundant triangle connecting to a concave corner (Figure 20(c)), the intersection region is enlarged and thereby the skeletal structure is distorted. The exterior triangle has a vertex at a concave corner and it conceals the concave corner. The last case is that triangles are generated in three edge segments or even three different edges. The shape shown in Figure 20d contains a thin crevice. The gray triangles are the redundant triangles split by the shape boundary (dot line).

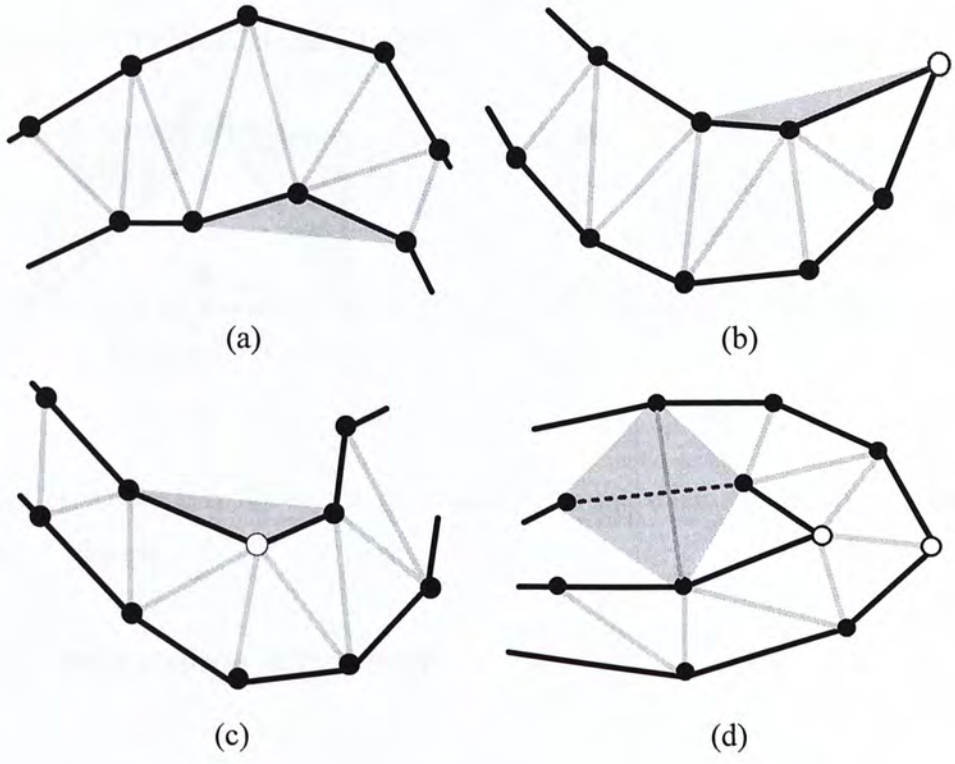


Figure 20. Configurations of redundant triangles. Triangles should be filtered with (a) vertexes in the same edge segment, (b) two vertexes in the same edge segment and the other vertex is a turning node, (c) two vertexes in different edge segments and one vertex in-between concave corner, and (d) triangles with three vertexes in different edge segments.

Some vacant regions may be induced in the triangle removal process. In Figure 21(a), a triangular vacancy is generated as two external edges and a missing edge (dot line). A triangle should be replaced to the region at shown in Figure 21(b). It first searches the triangles, which share a common isolated vertex (indicated with gray). The other vertexes are in the same edge segment and are extracted from the triangles. The three vertexes form an NT to replace the vacancy.

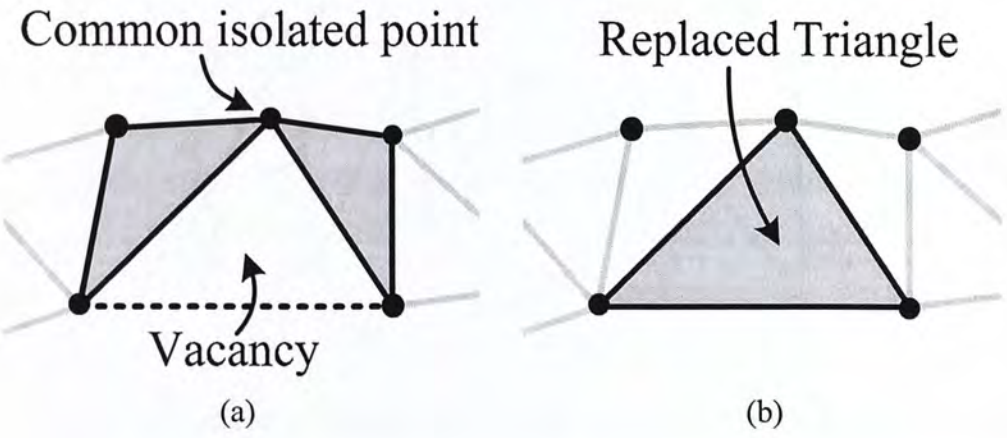
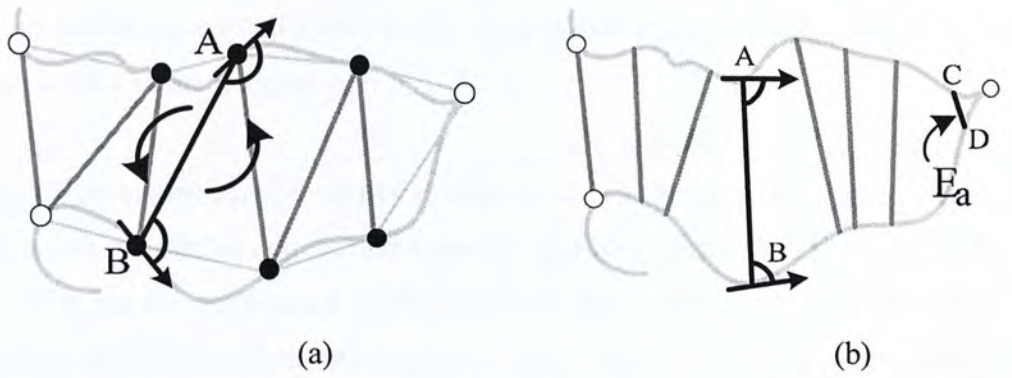


Figure 21. Triangle replacement: (a) vacancy in a shape and (b) the corresponding replaced triangle.

3.2.4. Internal Edge Refinement

The triangular mesh is enveloped by the boundary after replacing the misarranged triangles. Nevertheless, most of the internal edges on each triangular grid are not accorded with the SLS criterion, as shown in Figure 22(a). Their gradient vectors at the corresponding boundary points (arrows at the boundary) cannot form the same interior angles with the internal edge. So, the edge should rotate to minimize the difference in interior angles. If the angle at vertex A is larger than the angle at B, the edge rotates anti-clockwise, and vice versa. The angle differences can then be reduced and all the internal edge becomes SLS (Figure 22(b)). The mid-points of internal edges are extracted as a sampling point in the medial axis.

Moreover, the gradient of convex corners should also be evaluated for curve interpolation. This gradient is assumed to be perpendicular to the intersecting internal edge close to the corner. There exists two intersection points, as C and D in Figure 22(b), apart from the corner in different direction along the boundary. They are connected to form an additional internal edge, Ea. The convex corner then connects to the center of Ea as the terminal section of the skeleton.



(a) (b)

Figure 22. Illustration of internal edge refinement: (a) internal edge rotation and (b) the edge-refined pattern with an additional internal edge.

3.2.5. Skeletal Interpolation

The centers of the internal edges are only sampling points, and so they should be interpolated to obtain a connected skeleton. Previous research project applied the linear interpolation, which may cause obvious inconsistency with the target skeleton for insufficient sampling rate. Our approach adopts the Bezier curve because it can estimate a curve with small curvature changes. To preserve the gradient at the internal edge centers, the intervals between two edge centers are connected piecewise. The piecewise Bezier interpolation of skeleton is shown in Figure 23.

The skeleton branch is interpolated using two successive internal edges. The internal edge centers are the control points of the Bezier skeleton segment. The normal gradient of internal edge is the gradient of the symmetry point locus. If the gradients of the edge pair are not parallel, the gradient vectors from the corresponding internal edge centers would eventually intersect (Figure 23a). The intersection point, C_0, is another control point for the Bezier curve. A quadratic Bezier curve can be generated using (2). If the internal edges are parallel, two more control points should be artificially inserted as illustrated in Figure 23b. The control points, C_1 and C_2, are generated by shifting a margin from the internal

edge centers in the normal gradient directions (black arrows). In this case, a cubic Bezier is used as the interpolation.

The junction interpolation is shown in Figure 23(c). Arrows in the figure indicate the gradient of internal edge center P and the gradient of skeleton intersection C. Depending on the application, point C can be set as the center of intersection region or the characteristic skeleton point suggested in [19]. Since a smooth skeleton should be generated, two gradient vectors form the same interior angle with the line PC. A control point of Bezier curve can be generated by extracting the vector intersection C_3. Considering also points P and C, a quadratic Bezier curve can interpolate a line segment in the intersection region.

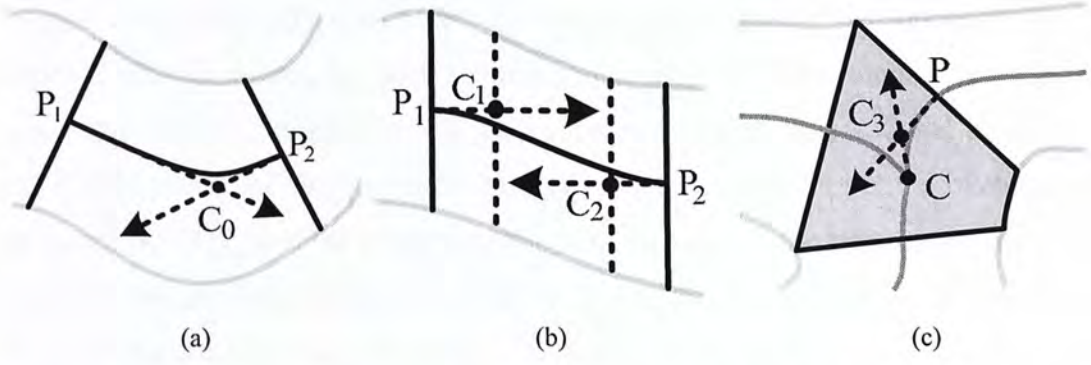


Figure 23. Bezier curve interpolation: internal edges of branch with (a) different gradients and (b) same gradients, and (c) junction interpolation.

3.3. EXPERIMENTS

To demonstrate the feasibility of the proposed method, an experiment was performed using a Chinese character (550x600 pixels) as shown in Figure 24(a). The character was segmented into triangles by DT. The misarranged triangles were modified to obtain a more precise segmentation. Afterwards, the external triangles were filtered by considering their centroids and vertex positions. Figure 24(b) shows the resultant image of the triangulation. Internal edges were refined

by rotating them to fulfill the SLS (Figure 24(c)). The internal edge centers were selected to be the sampling points in the locus of symmetry point. In the juncture regions, region centers were considered as the connection nodes. The locus was then interpolated using Bezier curve. The resultant skeleton shown in Figure 24(d) contains no spurious braches and scraggy curves.

Another experiment was conducted using a high resolution character image (500x400 pixels) to compare the proposed method with the other skeletonization methods, including ZSM's approach (ZSM) [19] and CDT [24]. Our method is a general method for patterns, especially for calligraphies and paintings. Here, we use an 'a' character as an example. The skeletons are indicated with red color in Figure 25 and the computational time is listed in Table 1. The proposed method (Figure 25(a)) and CDT with high edge sample rate (Figure 25(b)) obtained the better results. However, the edge sampling interval of the latter method was 1/5 times that of the proposed method. This induces a longer computational time in CDT (~36 times of the proposed method). Figure 25(c) shows the skeleton extracted by CDT with the same sampling rate the same as the proposed method. Although the processing time was slightly shorter, the skeleton is not precise. It was constructed with ragged curves. The result in Figure 25(d) is obtained by ZSM [19]. Although the skeleton is smooth and accurate, there are spurious branches at the tips indicated with arrows. The inaccuracy would potentially cause errors in the applications of pattern recognition. The processing time was ~3 times longer than the proposed method.

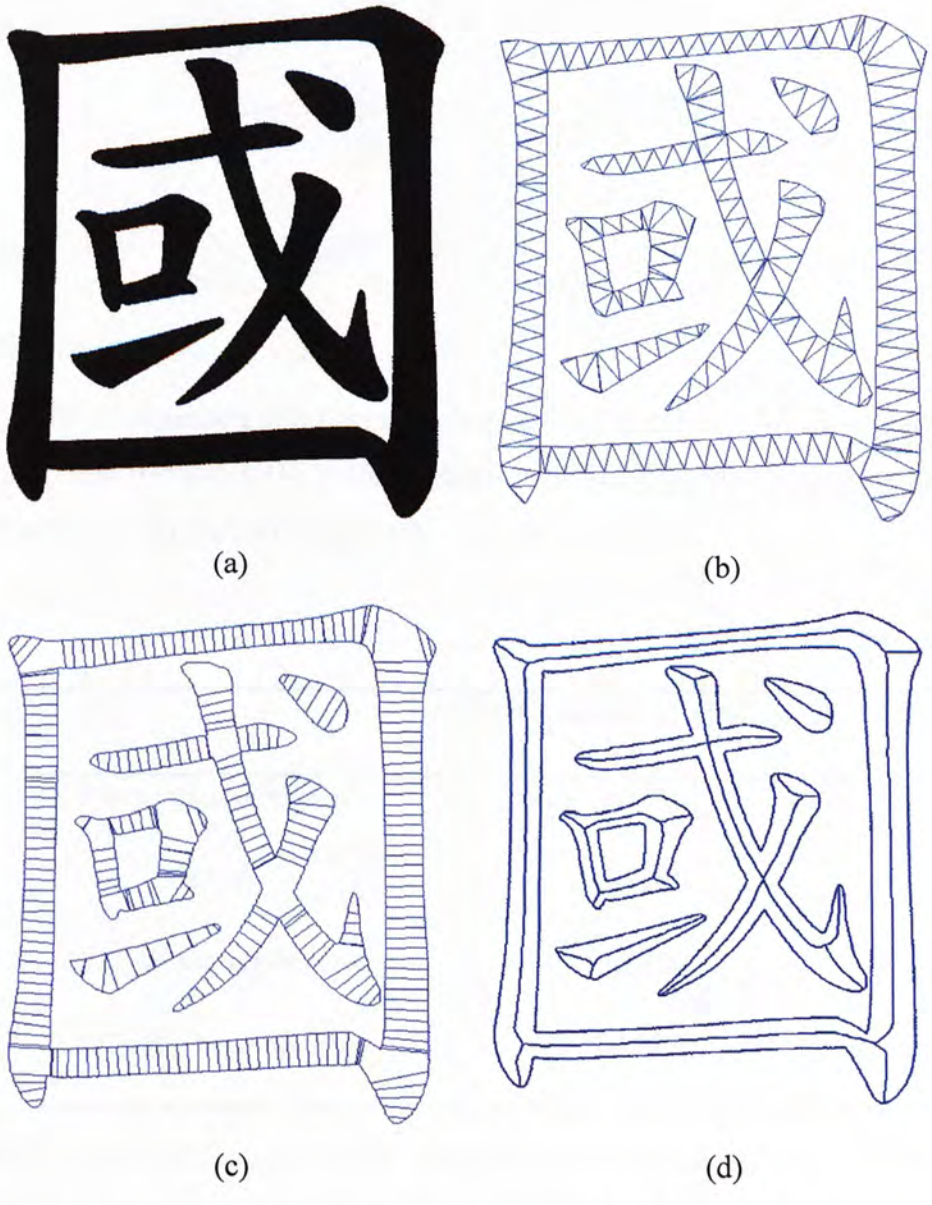
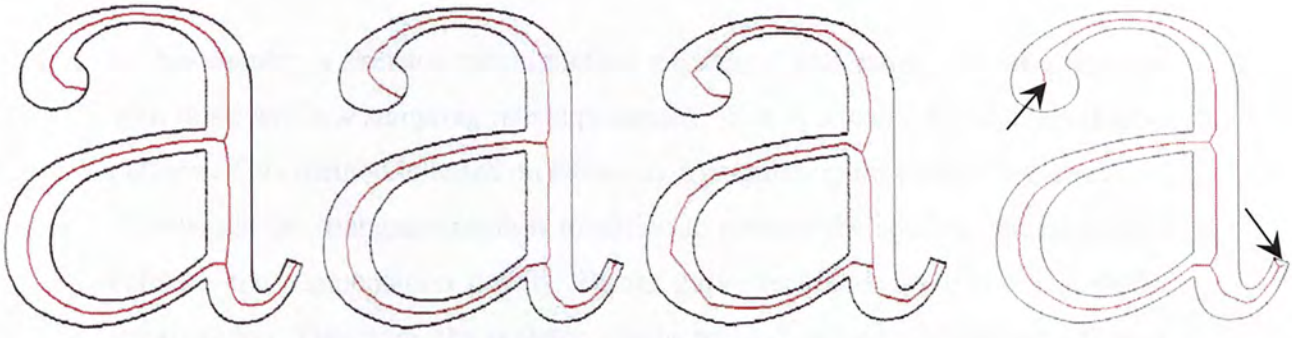


Figure 24. Experiment on a Chinese character: (a) the original pattern, patterns after (b) Delaunay triangulation, (c) internal edge refinement and (d) skeletonization



(a) (b) (c) (d)

Figure 25. Comparison of different skeletonization methods using character 'a': (a) the proposed method, CDT skeletonization with (b) high sample rate and (c) low sample rate, and (d) ZSM's approach

	Computational Time [sec]
The proposed method	1.4020
ZSM	5.9890
CDT (Low sample rate)	1.3420
CDT (High sample rate)	50.3750

Table 1. Computational time on 'a' using different skeletonization algorithms

3.4. CHAPTER SUMMARY

In this chapter, a skeletonization method capable of generating smooth skeletons with relatively low sampling rate is presented. It is applicable for high resolution patterns. This method is based on Delaunay triangulation for pattern segmentation. Afterwards the triangular mesh is modified to remove the spurious branches and enhance the segmentation quality. Bezier curve technique is used for skeleton interpolation. Therefore, the skeleton can be resized and scaled to adapt different applications. Moreover, the outputs of traditional methods are in the form of pixels, while the proposed method outputs the control points. This method can also be adopted in some applications requiring low memory storage, especially in mobile devices.

Experiment demonstrated that the proposed method can effective skeletonize shapes even with ribbon-like boundary. The proposed method was also compared with the different existing methods numerically. It shows that such method can perform efficient and high quality skeletonization. The computational time was ~35 times faster than the skeletonization using CDT, with the same quality of the output skeletons.

Chapter 4: STROKE SEGMENTATION FOR CHINESE WORDS

The main challenge for this problem is to reconstruct the connections of strokes, which are concealed by the stroke intersections. Traditional image segmentations are techniques to extract particular objects from a color image. So, they are not related to our application. Besides, stroke segmentation is a popular topic, which has been investigated by many research groups ([27]-[44]). They are focusing on segmentation for signature, which are not applicable to calligraphies. For these reasons, it is essential to have a new approach for Chinese calligraphy.

A new skeletonization method is proposed for fast medial axis generation in the chapter 3. It is obvious that the medial axis obtained is similar to stroke axes, the medial axis of each stroke, in a Chinese word. Many stroke segmentation algorithms for Chinese words ([38]-[44]) consider this axis to be the dominant information to determine the connectivity of stokes. However, the algorithms would have a higher potential to segment wrong strokes, as only the medial axis is considered to be the information to determine stokes. Some of them apply the skeletonization of pixel-removal approach, which would be hard to find the corresponding stroke segments of each skeleton branch, especially to words with large stroke width. Their computational time would also be due to the slow pixel-removal skeletonization.

In this chapter, we discuss the method for segmenting strokes from a Chinese word image, which uses some fundamental rules in structure of Chinese words and observations in general writing-styles of Chinese calligraphies. So, it would segment strokes correctly in most cases. This method is developed from the skeletonization method mentioned in the previous chapter. It is based on the refinement of the modified triangular mesh obtained from Delaunay triangulation and the suggested triangular modification.

4.1. RULE-BASED SPURIOUS BRANCHES REMOVAL

In the proposed skeletonization method, a pattern is triangulated into triangular mesh and it is split into branches with JT separated. Spurious branches (SBs), which have short medial axes, are the branches that caused by rugged pattern boundary that has relatively large gradient change. However, turning points (convex corners and concave corners) are determined by comparing the gradient change. This means if gradient change thresholding value is too large, some of the turning points would be neglected which causes omissions of some important strokes; otherwise, if the thresholding value is too small, there would be many spurious branches generated. To alleviate such problem, some of the convex corners are filtered.

In Figure 26, it shows some fundamental strokes for Chinese words. The convex corners are indicated with red circles, and the concave corners are highlighted with blue circles. It is obvious that Figure 26(a)-(k) are single strokes, while Figure 26(l) shows a compound stroke. The single strokes should contain only two convex corners, i.e. one for the starting terminal and the other for the ending terminal, the others are SBs, which are indicated by arrows in the figures. So, except Figure 26(a), the other strokes also have to remove some of the branches. SBs are short branches that bifurcate from the main medial axes. These bifurcations are not essential parts to describe the shape of the source pattern/word. They can be basically summarized into two general types: spurious branch in stroke terminal and spurious branch caused by turning stroke.

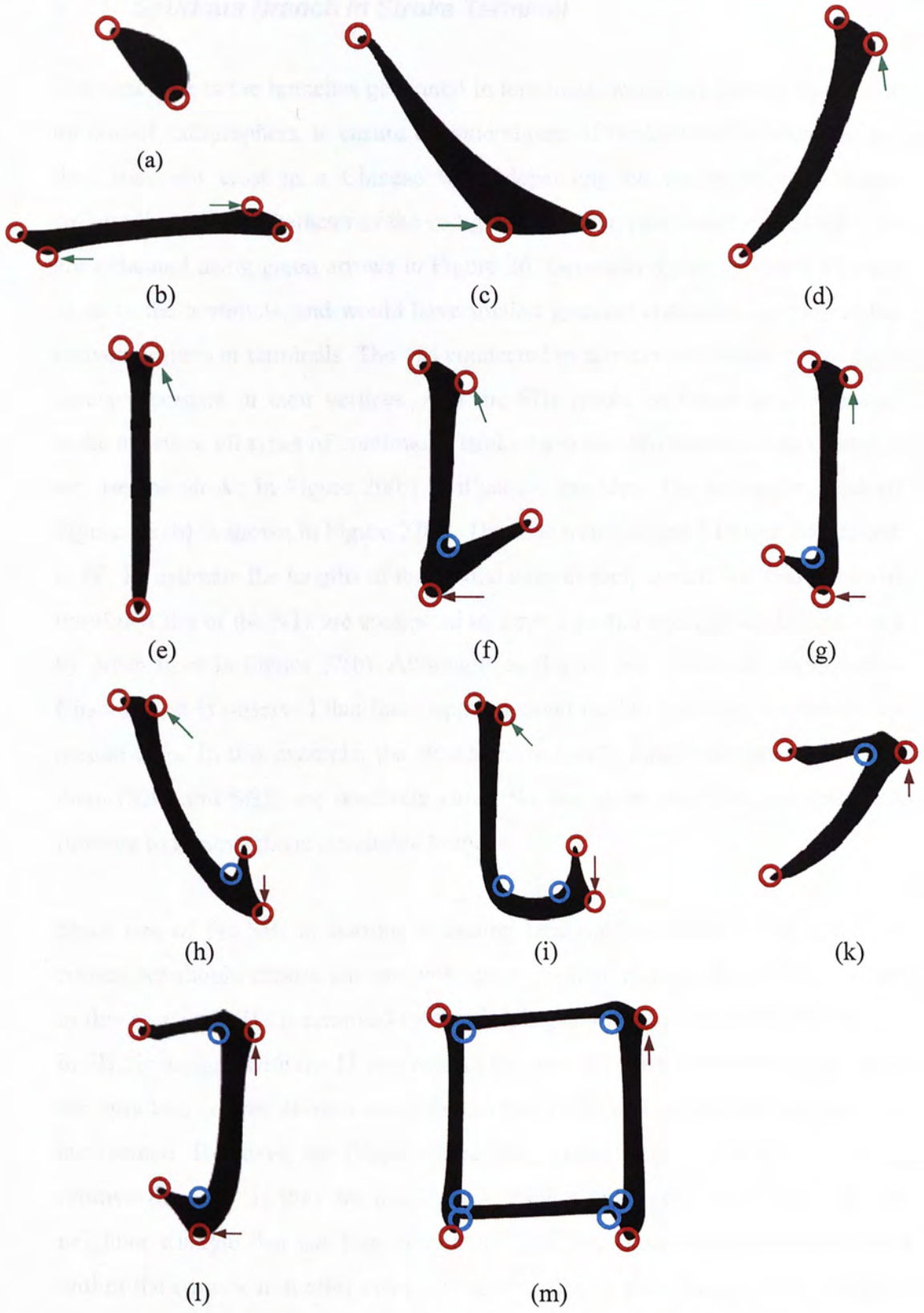
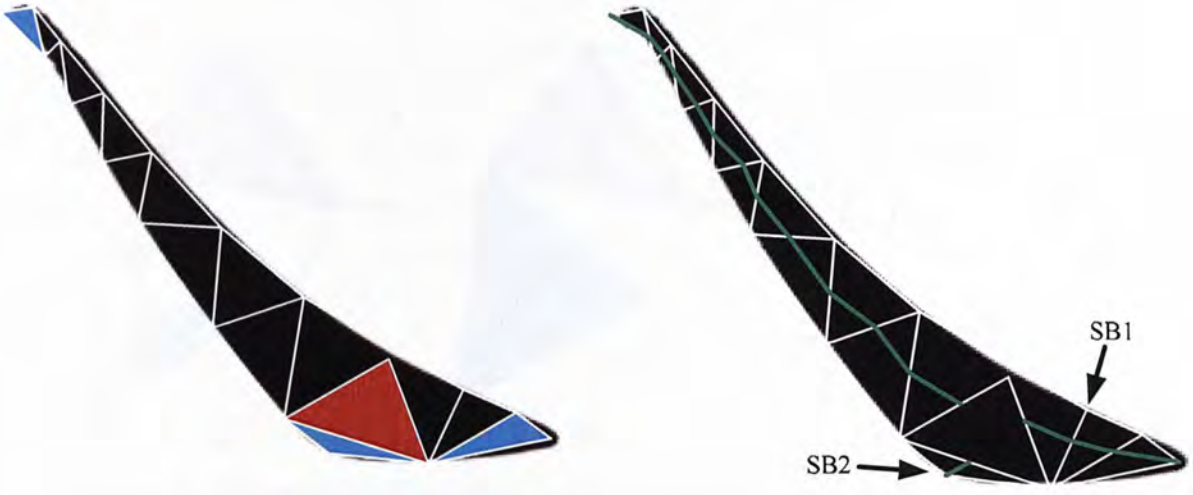


Figure 26. Fundamental strokes for Chinese words

4.1.1. Spurious Branch in Stroke Terminal

The first type is the branches generated in terminals, which are caused by special motion of calligraphers, to ensure the smoothness of the terminal boundaries. So, they may not exist in a Chinese word depending on the styles of Chinese calligraphy and the character of the calligrapher. The typical cases of this SB type are indicated using green arrows in Figure 26. Generally speaking, these SBs are close to the terminals, and would have smaller gradient change comparing to the convex corners at terminals. The JTs connected to this kind of SBs does not have concave corners in their vertices. And the SBs would be found in all types of stoke as, since all types of continuous stroke have two terminals. As an example, we use the stroke in Figure 26(b) to illustrate the idea. The triangular mesh of Figure 26 (b) is shown in Figure 27(a). The blue triangles are ETs and the red one is JT. To estimate the lengths of the medial axes in each branch, the mid-points of unrefined IEs of the NTs are connected to form a medial axes, which is illustrated by green lines in Figure 27(b). Although the IEs are not rotated as mentioned in Chapter 3, it is observed that these approximated medial axes can conjecture the medial axes. In this example, the strokes have totally three branches and two of them (SB1 and SB2) are relatively short. So, the short ones are undergone SB filtering to remove those unsuitable branches.

Since one of the SBs is starting or ending terminal, which often has a sharper corner, we should choose the one with larger gradient change. So, SB1 is selected in this case and SB2 is removed by the flipping algorithm. Therefore, the triangle in SB2 is merged with the JT and re-split into two NTs. As a SB is removed, there are only two convex corners remaining in the stroke and no medial axis branches are formed. However, the flipping algorithm, sometimes, may not guarantee to remove the SBs. If they do not form a convex hull (gray dot lines) with the neighbor triangle that has four vertices (Figure 28(a)), there is a vacancy found within the convex hull after merging the triangles. A new triangle fully occupies



(a) (b)

Figure 27. Example of spurious branch: (a) Triangular mesh of the stroke in Figure 26(b) and (b) linearly interpolated medial axes of branches

the convex hull is generated to replace the SB and its neighborhood as shown in Figure 28(b). This method would reduce the clarity of shape representationin that region due to the decrease in the number of boundary sample points. But, if the SB is removed, it would improve the overall quality of the approximated representation as the SB is found to be neglected information. After replacing the triangles, the SB may still exist if the SB contains more than one triangle. Then, the triangular branch is flipped using the flipping algorithm to entirely remove the branch. Similar to the flipping algorithm applied in the triangle modification in the proposed skeletonization method, the number of flipping is equal to the number of triangles in the branch. The reason is each flipping reduces one triangle in the branch. In Figure 28(b), two triangles remain in the branch, so two flipping is used to eliminate them. And in the first flipping, the NT connected with the JT (red) is flipped. As the SB reduce one triangle, another branch connected to the same JT gain one triangle. Therefore, this would shift the JT to the end of the SB, and finally both the JT and the SB disappear. The flipped triangles, shown in Figure 28(c), are indicated with yellow color.

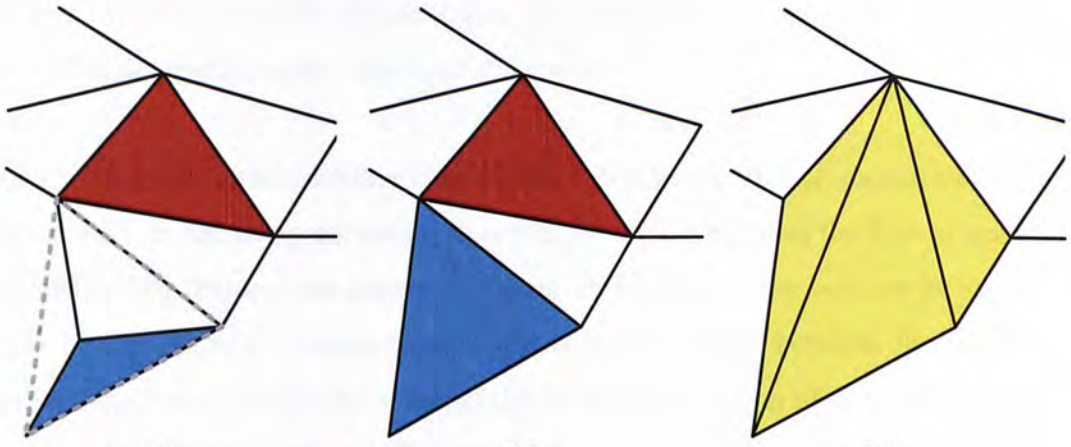


Figure 28. Illustration of spurious branch removal: (a) non-removed spurious branch, (b) triangle replaced mesh and (c) combined and flipped triangles

4.1.2. Spurious Branch Caused by Turning Stroke

The other type of SB is caused by the turning motion, which is abrupt direction change, in stroke trajectory. In other words, it originally exists in the structures of Chinese words. This kind of SB can be found in the strokes shown in Figure 26 (f)-(m). They are illustrated using brown arrows. Comparing to the SB in stroke terminals, this type of SB is more complicated to be classified. Since it has many possibilities to case this SB type, they are generalized into three cases.

The first case is shown in Figure 26 (f). It comprises a vertical stroke and a stroke which is nearly horizontal as that in Figure 29(a). The SB is generated when the calligrapher is turning the brush to draw the inclined stroke after drawing the vertical one. Also, in the triangular mesh, the JT often has a concave corner due to the change is stroke direction. To search this SB, we consider the branches connected the same JT. Apart from the SB, there are two branches split by the JT. One is a straight stroke branch which form an angle around 75 -105 deg to the horizon, and the other stroke leans about 15-45 deg to the horizon. The lengths of both stroke branches are not significant to decide if SB exists, because this stroke

appears in different words. The positions of the stroke and lengths of the branches are varied depending on the structure of the word.

Figure 26 (g)-(k) shows another type of SB. It has a short branch connected with the same JT in the triangular mesh. However, it is different from the SBs in stroke terminals. The features are shown in Figure 29(b). One of the vertices in the JT (red) of the SB is a concave corner, which is not located between the convex corners. Referring to the Figure 26 (g)-(k), the angle direction of the corner in the short branch should have a reference range, since it can be observed that the angles are pointing upwards. However, in Chinese calligraphy, the angles may be varied due to the style and creativity of the calligrapher. Therefore, it is not limited by a specific range.

Figure 26 (k)-(m) is the third case. The characteristic of these strokes is they have an ET with a convex corner pointing to the right side. An example is shown in Figure 29(c). Since the basic rule for Chinese word states that strokes are written from the top to the bottom and from the left to the right, the stroke should be written as the direction illustrated in Figure 29(c). Therefore, a horizontal stroke branch is connected to the top edge of the JT and another one with medial axis gradient pointing down the from the bottom edge of the JT. For the SB, the angle direction is supposed to be within -90 to 135 anticlockwise to the horizontal axis.

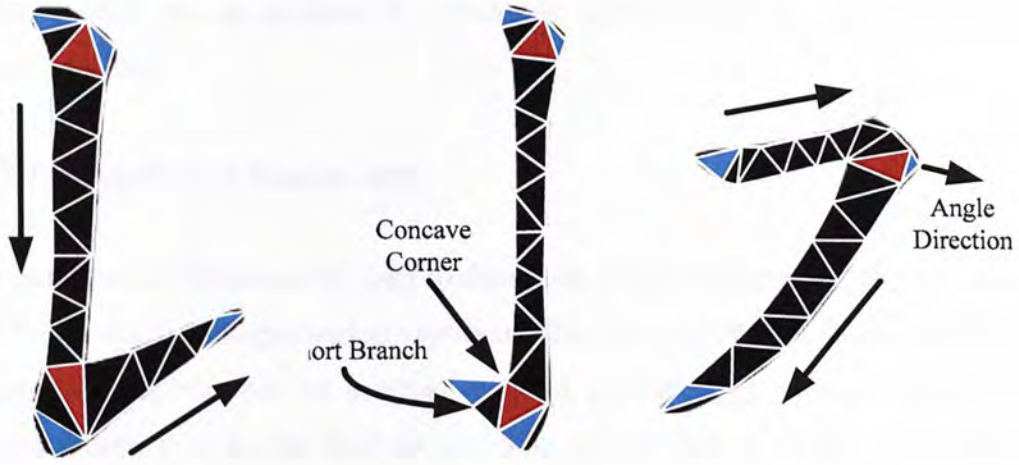


Figure 29. Three General cases of spurious branch in turning stroke

The SB removal is applied to Chinese word to find out SBs. When all the SBs are classified, the triangular mesh is conducted using internal edge refinement as mentioned in Chapter 3. So, the mesh is converted to be internal edges. It is a clear shape representation, since the internal edges are cross-section lines that cutting the boundary. They contain the information of medial axes and the gradients of boundary at the example points.

4.2. STROKE CONNECTIVITY DETERMINATION

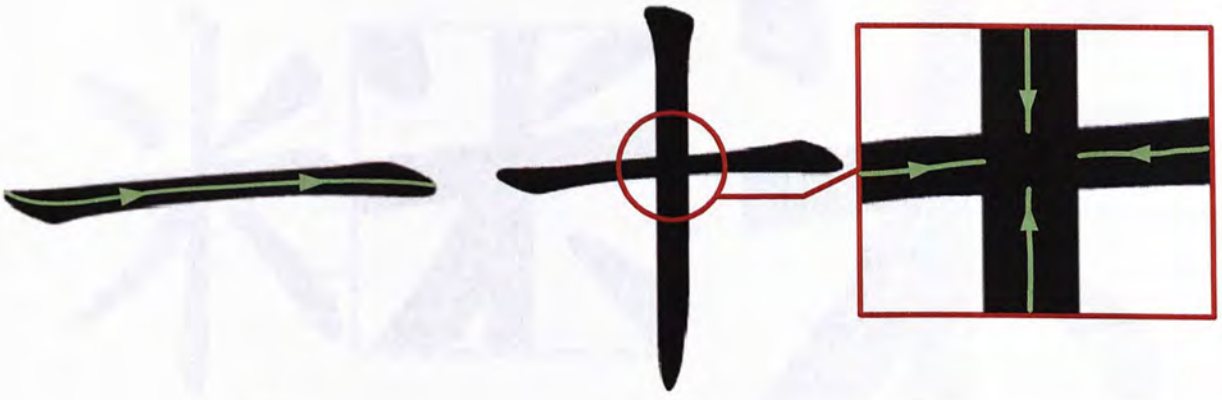
The previous section shows how to modify the triangle mesh of the fundamental strokes to be desired shape representation. However, in a Chinese word, the strokes intersect each other to form a complicated structure. The intersecting regions separate a word into individual branches, which signifies that strokes are not apparent to be obtained. To fully segment strokes from a word, it is required to consider the connectivity between the branches and choose those with larger tendency of connection for merging. Therefore, it is an optimization problem to search the branches with specific relationships. In other words, the branches have relatively small differences in some factors. The factors for stroke connectivity

determination include gradient of medial axis, gradient of branch boundaries and branch widths.

4.2.1. Gradient of Medial Axis

As mentioned, medial axis is used to determine stroke connectivity, since it is an essential way for morphological representation. Referring to the strokes show in Figure 26, if branches in a stroke are not generated by turning motion of calligraphers, their medial axes are normally with limited gradient change. Also, considering the structures of Chinese words, it seldom has strokes intersecting in the stroke turning part, which would generate second type of SB. Therefore, those branches with similar gradients and connected to the same intersection region are classified to be components of a stroke.

An example in Figure 30 is shown as illustration. Figure 30(a) is a horizontal stroke, which is one of the fundamental strokes in Chinese words. The green line is its medial axis and the arrows indicate the gradient direction. It is a horizontal line which indicates the stroke direction. A Chinese word with a horizontal stroke is shown in Figure 30(b). In the red box of the figure, the intersection region displays as a cross shape. Although the connectivity of the horizontal stroke is blocked by the intersection region, it can also be observed that the word comprises two strokes, a vertical stroke and a horizontal stroke. The reason is there are two pairs of medial axes, and each of the pair has two branches with similar gradient.



(a) (b)

Figure 30. Illustration of stroke connectivity determination based on medial axis gradient: (a) a horizontal stroke and (b) a Chinese word with the stroke in Figure 30(a).

However, some branches would be incorrectly matched, if the gradients are directly compared. A case is shown in Figure 31(a). The medial axis directions of two branches are pointed with green arrows. The branches have close gradients and extend from the same intersection region, they, however, are two separated strokes. Hence, another method, which is illustrated in Figure 31(b), is used to compare the gradients of branches. A reference line, indicated in yellow, connects the terminals of the branch medial axes. The gradient difference is the summation of the angles (blue) between the reference line and the medial axes. After drawing the reference line, it is obvious that the branches do not belong to the same stroke, since there is spacing between the medial axes. If we extend the axes in the arrow directions, they do not have an intercept point in short distance.

(a) (b)

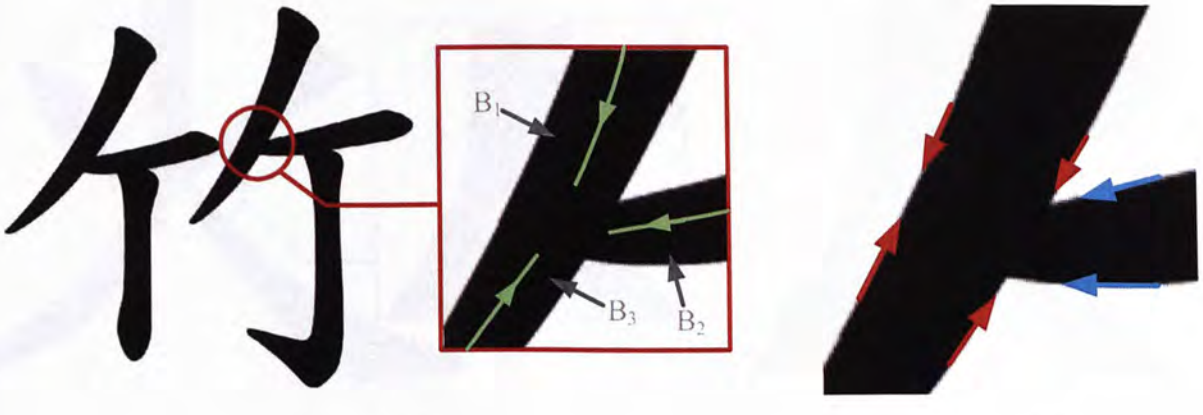
Figure 31. Comparing method of medial axes gradient: (a) example of unmatched branch and (b) illustration of the method

4.2.2. Gradient of Branch Boundary

Except gradient of medial axis, gradient of branch boundary is also considered to be utilized for branch matching. After the triangle modification and internal edge refinement, a word is split into many branch segments. Branch boundary is part of their boundaries that line on the contour of the original word. It is also boundary segments of the original word. Therefore, if the branches, connected to the same JT, belong to the same stroke, their branch boundaries are also part of the stroke boundary. Since the branch boundaries are only separate by the intersection region, which is often a short distance, the gradient change would not be too large. The advantage of adopting this factor is that it can also be applied to branches with imprecise shape representation. Stroke matching which only uses gradients of medial axes requires high accuracy in founding the axes. However, uncertainties may exist when rotate the internal edges in the triangular mesh, due to the boundaries are not perfectly smooth. This problem is more critical, if the word is low resolution or it is written by calligraphists (not computer fonts). So, branch boundary provides additional information alleviate the problem caused by the error in medial axis.

This method is shown in Figure 32. Figure 32(a) is an example showing a case that may cause mistake to find the corresponding branches. An intersection region in the left side of the word has three branches, and the green arrows indicate the gradients of each branch. The bottom branch, B_3, has similar gradient differences to the other branches (B_1 and B_2), which is equivocal to match them. In contrast, if we consider the branch boundaries, it would be more obvious to determine which branches are actually connected. In Figure 32(b), the branches of the intersection region are indicated with arrows to show the directions of branch boundaries. The red arrows are the gradients of branch boundaries in the same stroke, while the blue arrows are the boundary directions of the isolated branch.

To find the gradient difference between the boundaries, the scheme suggested to comparing medial axes is used. The reason is that if a continuous stroke boundary is separated by an intersecting branch, like the one in Figure 32(b), there is a short distance between the boundary segments, which would induce gradient difference in the segments. If a boundary segment in another branch has similar gradient which would give a smaller difference in gradient, incorrect stroke connectivity may occur.



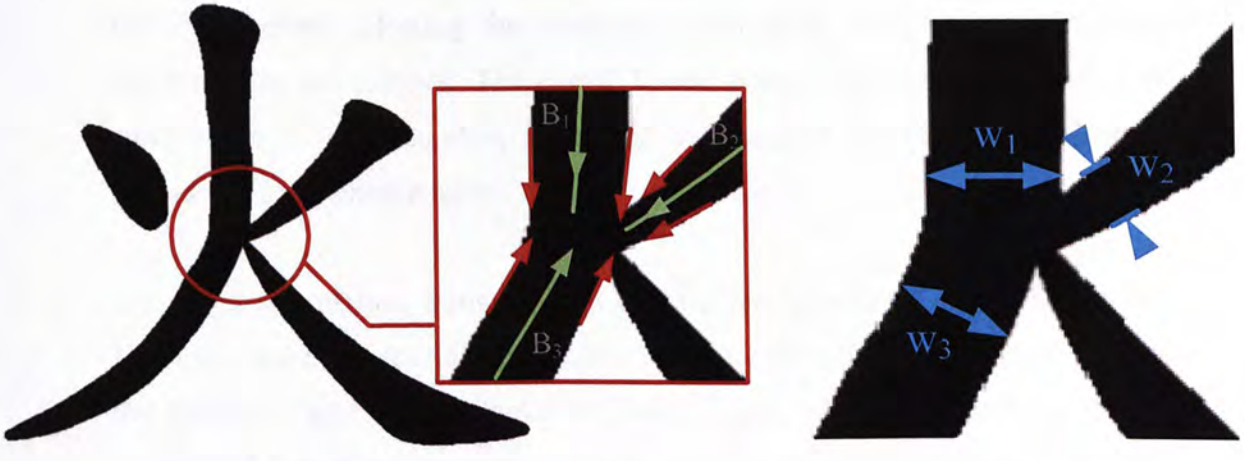
(a) (b)

Figure 32. Figure showing branch matching using gradient difference of branch boundaries: (a) a Chinese word with an indicated intersection region and (b) the directions of its branch boundaries

4.2.3. Branch Width

In addition to the bases above, branch width is another factor to affect the stroke connectivity. Imagine that, in an intersection region, if two branches belong to a stroke, the width would not change much. Similar to the case for gradient of branch boundary, stroke width is often uniform in a short range. Although the gradients of medial axes and branch boundaries can solve most, branch width would improve the contrast in the rating of determining which branches should be connected.

Figure 33 shows an example which would be ambiguous to decide the corresponding branch pair. Figure 33(a) indicates that the gradients of medial axes (green) and branch boundaries (red) in each branch. It is found that no matter comparing the medial axes or branch boundaries of B_3 to that of B_1 and B_2, it would also give similar gradient differences. According to Figure 33(b), if branches widths are compared, the result is clearer, as B_1 and B_3 closer width (w_1 and w_3) and B_2 has a relatively small width (w_2).



(a) (b)

Figure 33. Comparing branch widths for stroke matching: (a) and (b) widths of each branch.

4.2.4. Combined Objective Function

After combining the three factors above, the objective function F_{obj} for branches matching can be written as:

$$F_{obj} = \omega_1 \Delta G + \omega_2 \Delta B + \omega_3 \Delta W \qquad (2)$$

where ΔG is the gradient difference in medial axes, ΔB is the gradient difference of the boundaries of branches and ΔW is the width difference, which is a fraction (larger width / smaller width). ω_1, ω_2 and ω_3 are the weighting for ΔG, ΔB and ΔW respectively. ω_1 and ω_2 should have the larger values, because the gradient differences are dominant to show the relationship between branches. ω_3 is comparatively small, since it is adopted as auxiliary.

Then, the objective function is applied to every two branches in the same intersection. When all the scores are derived, we may match those with low scores. In some cases, the branches do not have correspondence, e.g. a stroke only intercepts other strokes at its terminal, though they are intersected. These branches have high scores, which would potentially connect unrelated branches into strokes. To prevent the incorrect merging, a thresholding value is used to filter those branches before selecting the branches. Afterwards, the remaining branches combinations are selected. Those with lower scores have higher priorities. And, since there is no redundant matching, the selected branches would not be considered to be chosen again.

The proposed method, basically, can find the connectivity of almost all strokes. However, there is a special case, shown in Figure 34, would not be determined by the gradients and stroke thickness. Both Figure 34(a) and Figure 34(b) are recognized to be the same word. The difference is the red stroke is attached to the others in Figure 34(b), while that in Figure 34(a) does not. Therefore, if only the proposed method is used, that in Figure 34(a) would be segmented to be one stroke and the one in Figure 34(b) would be broken into two. They should be matched after using the objective function. If in the same intersection region a

horizontal stroke and a inclined stroke (in the direction pointed by the arrow) that are not matched with other branches, they are connected to form the stroke shown in Figure 26(k).

(a) (b)

Figure 34. A special case for determination of stroke connectivity: (a) and (b) are the same word in different forms.

4.3. STROKE GENERATION

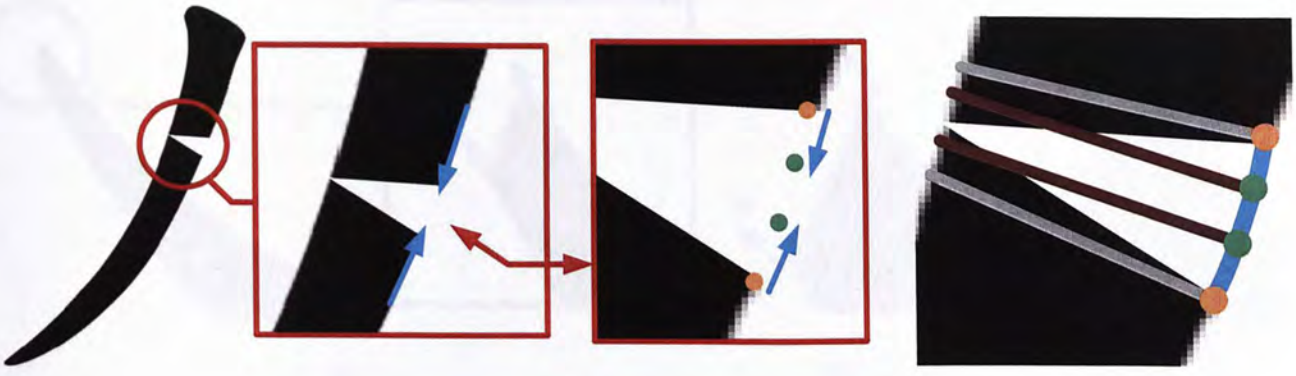
After determining the connectivity of strokes, the connected branches should be extracted as full strokes. Since strokes would have some overlapped regions that cannot be directly segment, the regions are generated to retain the stroke continuity. There are two cases of the stroke generation. One of them is in stroke terminal and the other is between branches. To ensure the smoothness of stroke boundary, Bezier curve is utilized for boundary interpolation. Figure 35(a) shows a Chinese word comprises strokes that are extracted using both types of stroke regeneration. Its segmented branches are displayed in Figure 35(b_1), (b_2) and (b_3). It is apparent that strokes in Figure 35(b_1) and (b_2) belong to the same stroke, while the one in Figure 35(b_3) is another stroke.

(a) (b_1) (b_2) (b_3)

Figure 35. (a) A Chinese word with branches that should be created using both stroke regeneration methods and (b) its segmented branches.

4.3.1. Stroke Connection between Branches

We first discuss stroke generation between branches. The aim of it is to complete a stroke. Consider the example in Figure 36, two branches are verified to be connected. There is a crevice between them as shown in Figure 36(a). The blue arrows indicate the directions of both boundary segments, which infer that the segments should be extended in the directions, such that they are intercepted at the same point. The interpolation is illustrated in Figure 36(b). The colored circles are control points for Bezier curve interpolation. The orange circles are the end points of the boundary segments, while the green ones are points shifted from the orange circles in the boundary directions. Since it provides four control points, the curve is quadratic, which can be derived using equation (1). The generated boundary is shown in Figure 36(c). The lines in the figure are cross-section lines of the stroke. The gray lines are cross-section lines of both branches and the brown lines are generated for ensuring smooth medial axis of the stroke. So, the whole shape representation of the stroke can be obtained by combining the cross-section lines of the branches and those created.

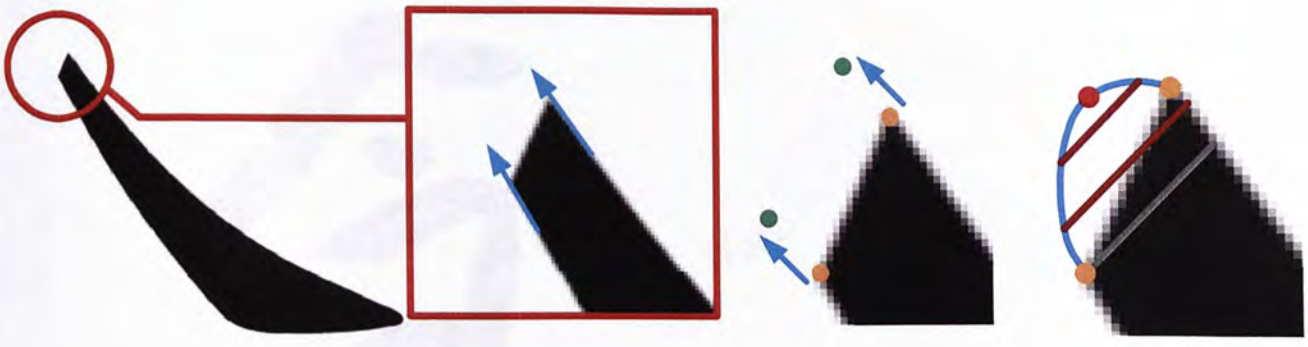


(a) (b) (c)

Figure 36. Stroke connection between two corresponding branches: (a) the intersection region of the stroke, (b) control points for boundary interpolation and (c) an interpolated boundary

4.3.2. Stroke Generation in Stroke Terminal

A branch with stroke terminal is an isolated branch, so it is not connected with other branches. The missing part of the stroke is generated using the information of the branch. Figure 37 shows an example of an isolated branch. The blue arrows (Figure 37(a)) are the directions of the branch boundaries on both sides. Figure 37 (b) shows the control points of the Bezier curve. The orange circles are at the branch boundaries. Similar to stroke generation between branches, two control points are extended from the directions of the branch boundaries. The interpolated stroke terminal is shown Figure 37(c). The red point is the mid-point of the interpolated curve, which is also the end of medial axis. To obtain the shape representation, the cross-section line of the generated part should be added to the stroke representation. The gray is a cross-section line of the branch, the brawn lines are generated. They are searched along the curve by shifting the points of the gray line with a constant distance.



(a) (b) (c)

Figure 37. Stroke generation in stroke terminal: (a) the terminal to be generated, (b) control points

4.4. Experiment Using Intelligent Robotic Art System

To examine the proposed stroke segmentation method, it is used to extract the strokes in one of Bada Shanren's calligraphies. The segmentations of words are shown in Appendix. The result shows that the proposed method can segment most of the intersection region.

One of the words shown in Figure 38 is discussed in detail. Figure 38(a) is the original (binary) word. It is first triangulated using Delaunay triangulation and becomes a triangular mesh. The mesh is first modified, like the process in the proposed skeletonization method, to group the JTs into an intersection region and remove the inappropriate ETs. The further triangle modification is applied, following the rules suggested based on the structure of Chinese words, to eliminate spurious branches. The mesh after the modification is shown in Figure 38(b). The internal edges of the mesh are rotated to accord with the SLS (Figure 38(c)). Since spurious branches are removed, the remaining branches are merged to form full strokes. Their connections are determined using equation (2), which is

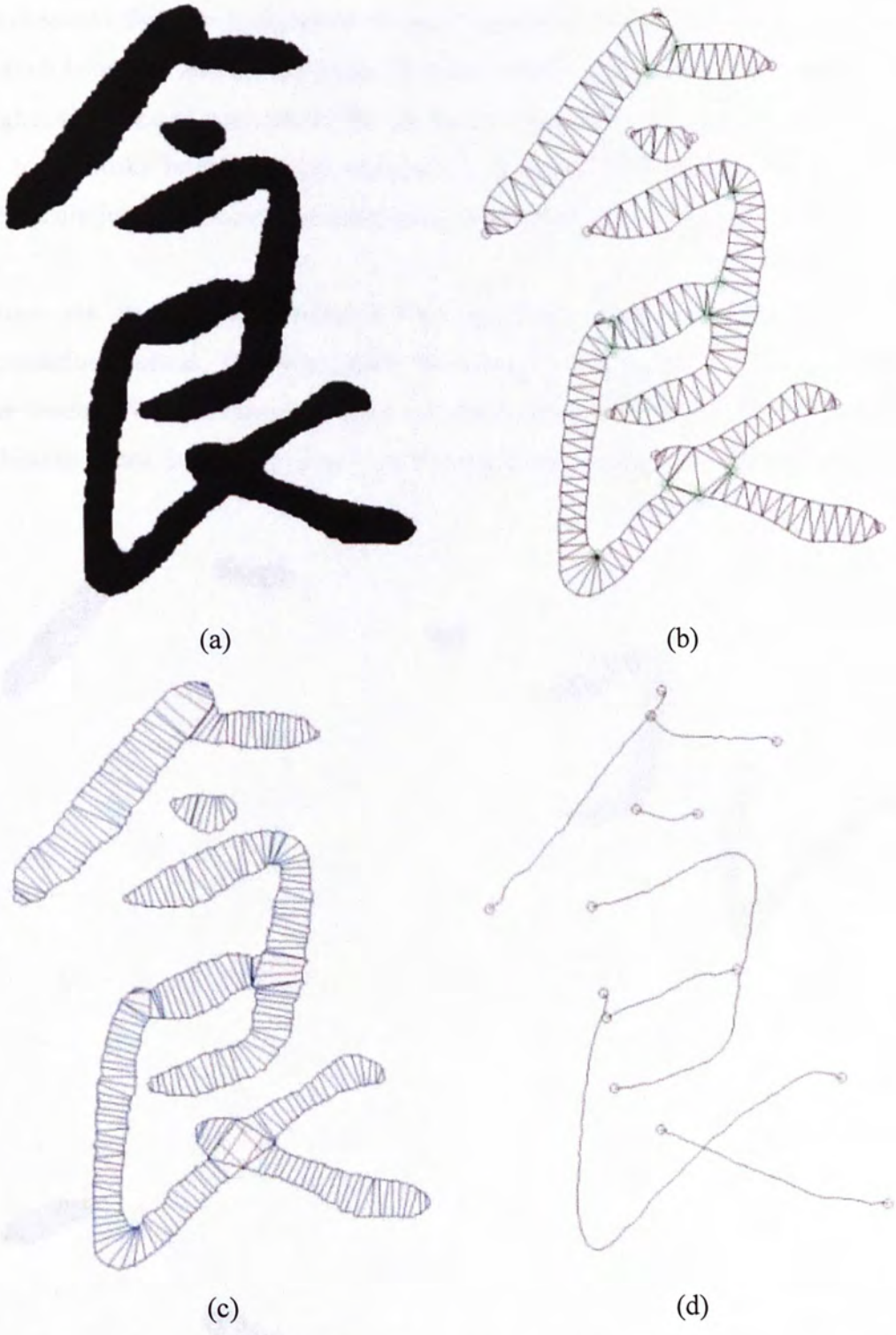
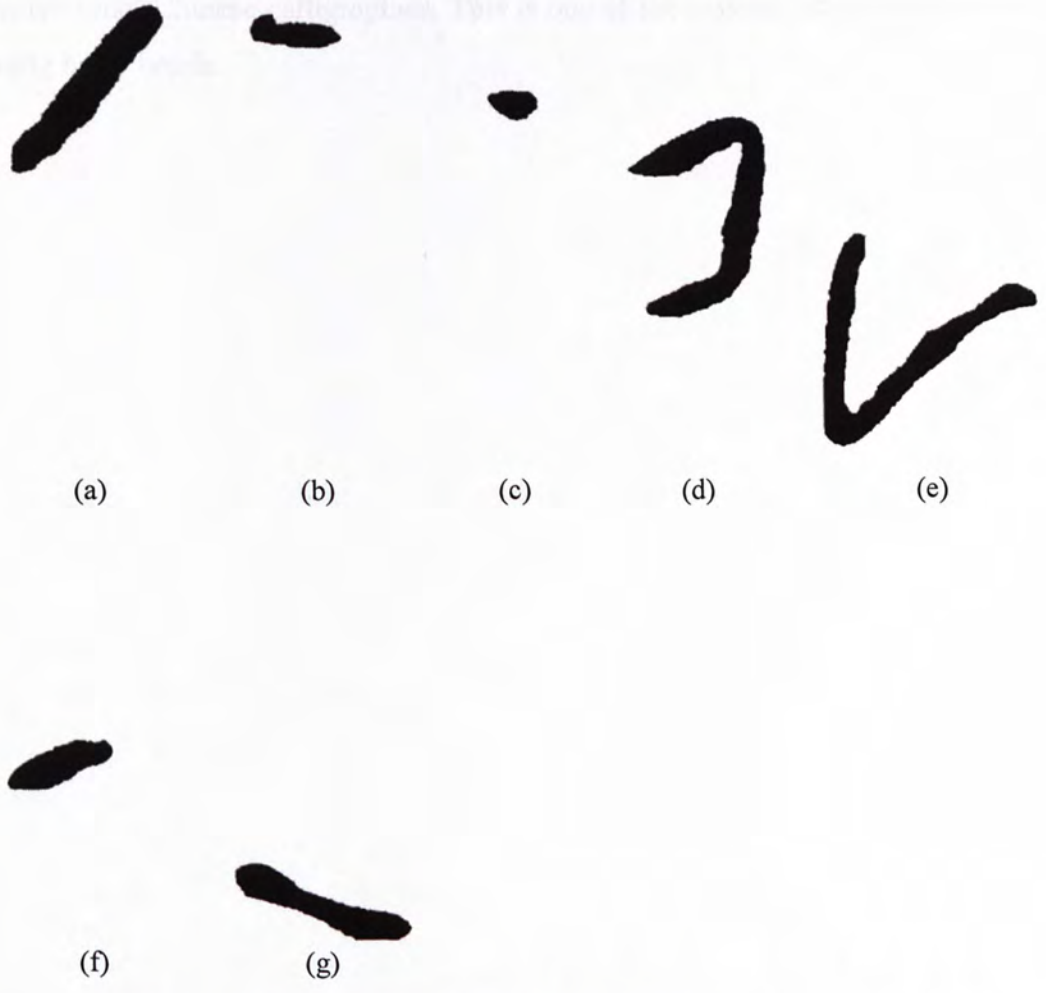


Figure 38. A Chinese word analyzed using the proposed stroke segmentation method: (a) original word, its triangular mesh, refined internal edges (cross-section lines) and the medial axes of the segmented strokes.

an objective function includes the score of gradient differences of medial axes and branch boundary and the difference in branch width. The lower score indicates the higher tendency of connection. So, the branch pairs with lower scores are merged to form stroke branches. The mid-points of refined internal edges in the same stroke are joined to form the medial axes of a stroke as shown in Figure 38(d).

Then, the strokes are generated from the stroke branches using the stroke generation method. The boundaries are created using Bezier curve, so it retains the boundary smoothness. Figure 39 shows the segmented strokes from the Chinese word in Figure 38(a). Afterwards, the segmented strokes are then

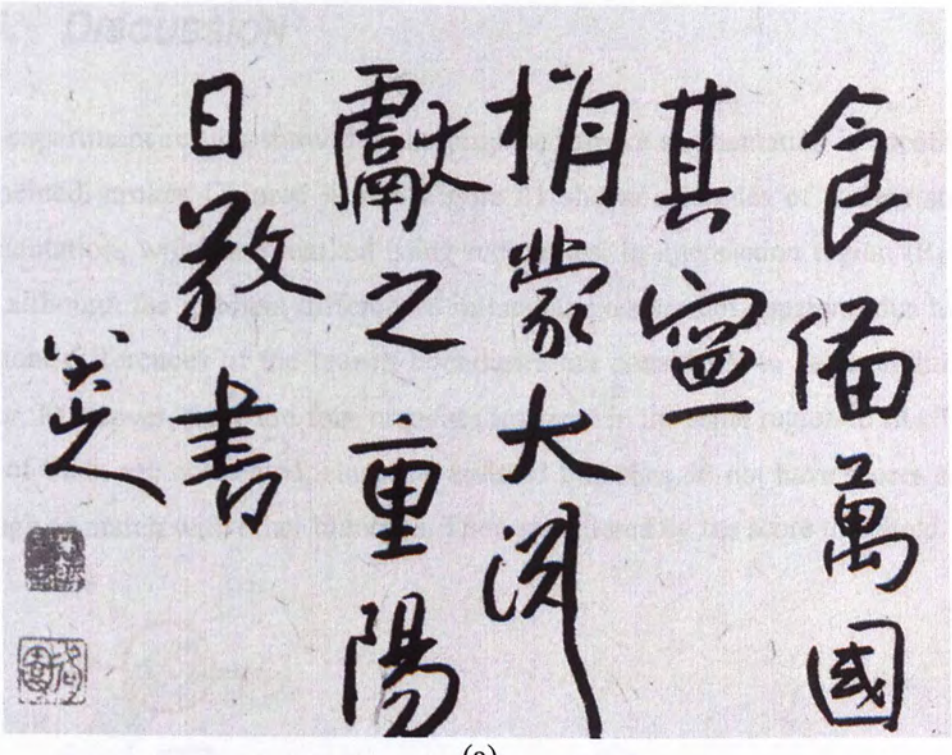


(a) (b) (c) (d) (e)

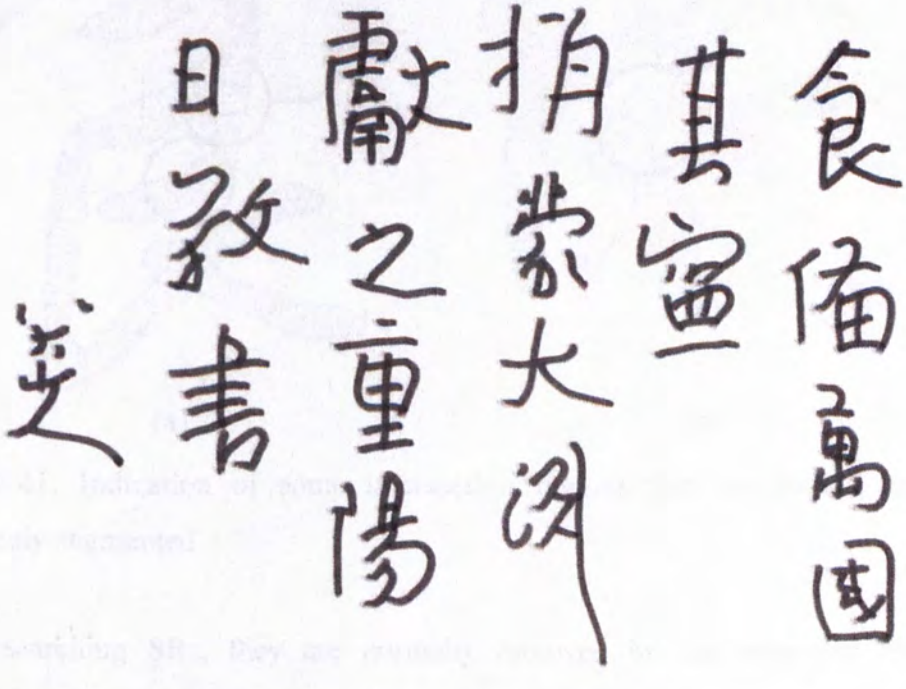
(f) (g)

Figure 39. The segmented strokes of the Chinese word in Figure 38(a)

executed using IRAS. In this experiment, since it is a preliminary test, only the x, y and z axes are activated, and the other rotational joints are disabled. A non-hairy brush is used to write the words with constant brush insertion into the paper, the output, therefore, would have constant thickness. The original Bada Shanren's calligraphy is in Figure 40(a). As discussed above, the words are segmented. The medial axes of the strokes are converted into commands and feed into IRAS for execution. Figure 40(b) shows the mechanical output written by IRAS. The result can approximately emulate the shapes of words. However, one of the essences in Chinese calligraphy is the varying stroke thickness. Constant thickness would reduce the aesthetic feeling of the calligraphy. Normal non-hairy brush can only generate strokes with limited thickness, which is not sufficient for performing professional Chinese calligraphies. This is one of the reasons for the necessity of using hairy brush.



(a)



(b)

Figure 40. Calligraphy emulation: (a) Bada Shanren's calligraphy and (b) emulated version with constant stroke width.

4.5. DISCUSSION

The experiment results show that the proposed stroke segmentation is capable to segmented strokes Chinese words. Figure 41 shows examples of correct stroke segmentation, which are marked using red circles. In intersection region IR_1 and IR_2, although the gradient differences in medial axes are not apparent, due to the gradient differences of the branch boundaries are considered to be an additional factor. Moreover, there are four branches intersect in the same region in IR_2. Only two of them are connected, since the isolated branches do not have scores small enough to match with other branches. They are filtered by the score threshold.

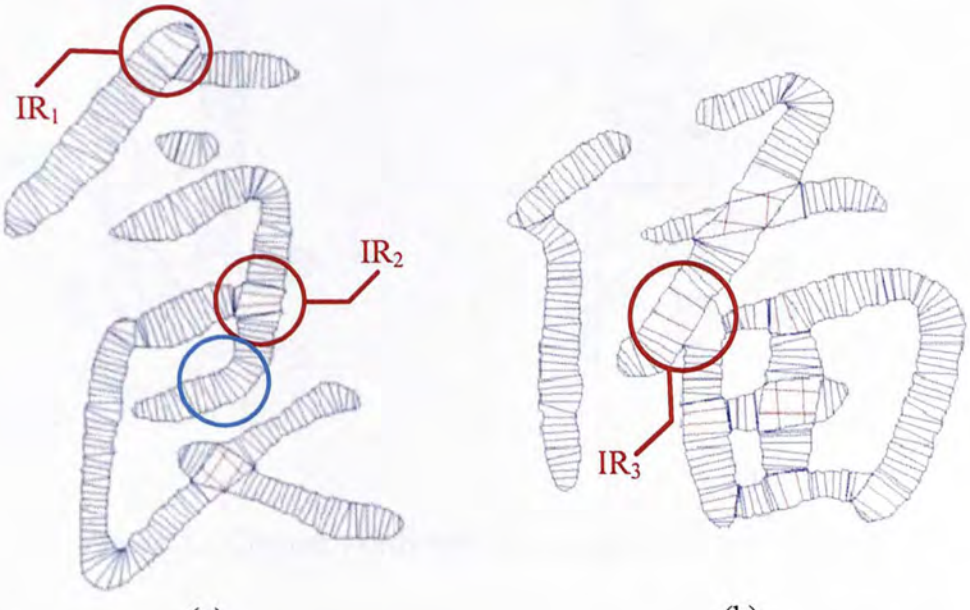


Figure 41. Indication of some intersection regions that the strokes can be accurately segmented

After searching SBs, they are normally removed by the suggested triangle modification method. However, there are still some situations that cannot be distinguished. They are shown in Figure 42. In Figure 42(a), the intersection region in the blue circle is connected with four branches. The stroke1 and stroke 2 are identified to be in the same stroke, since the gradient change of the medial axis

is small, the other factors does not strong enough to show the branches are actually disconnected. Besides, the blue circles in Figure 41(a) and Figure 42(b) highlight branches, which should be split into two separate branches. However, the gradient changes of the branch boundaries are not large to be recognized, and no convex corners are found. Therefore, Intersection regions are not formed to break the branches.

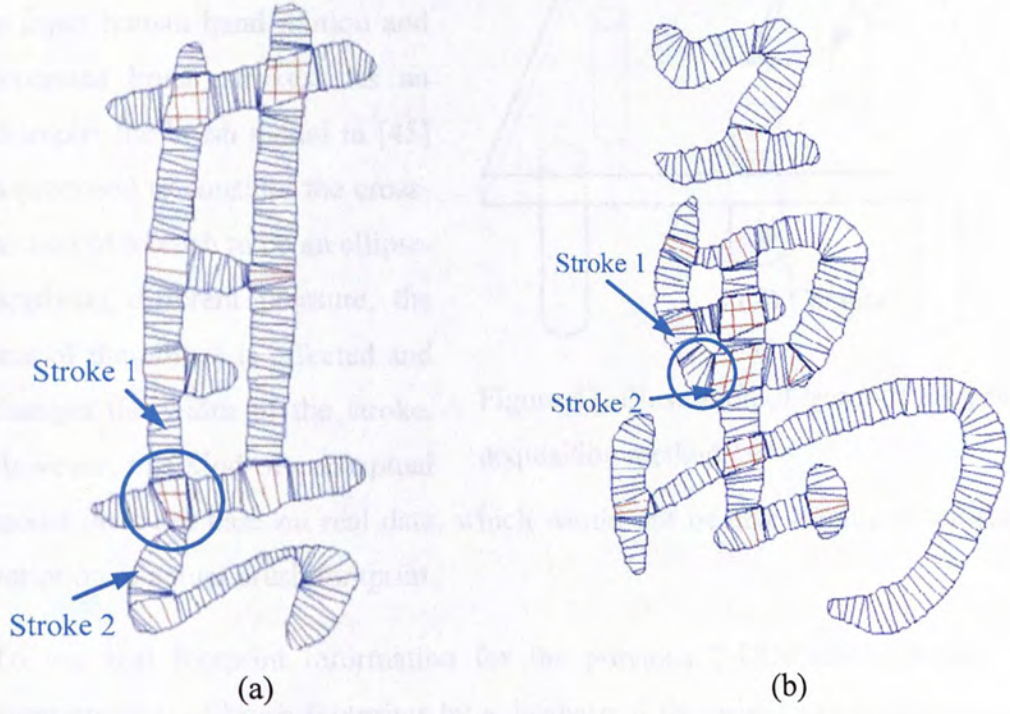


Figure 42. Chinese words with inaccurate stroke segmentation

Chapter 5: EXPERIMENTAL ACQUISITION OF BRUSH FOOTPRINTS

Typical brush footprint models are used as a virtual brush ([45]-[49]). It works with input devices to input human hand motion and generates brush strokes. As an example, the brush model in [45] is proposed to consider the cross-section of a brush to be an ellipse. Applying different pressure, the size of the ellipse is affected and changes the width of the stroke. However, this kind of conceptual model does not base on real data, which would not be able to predict the shape variation of actual brush footprint.

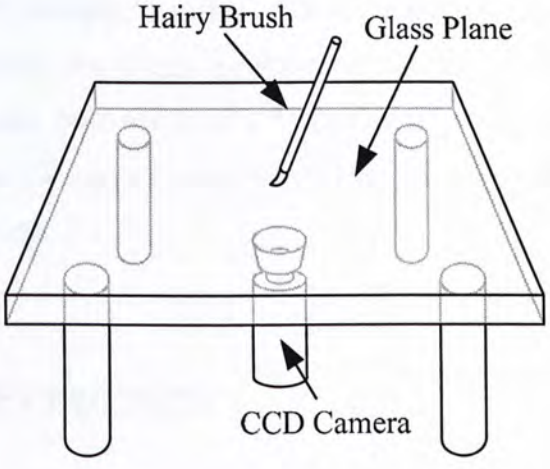


Figure 43. Illustration of pervious footprint acquisition method.

To use real footprint information for the previous 3-DOF configuration, the representation of brush footprints by a database of footprint images was proposed to acquire a model. The conceptual idea is illustrated in Figure 43. A brush is moving and contacting on a glass surface with different pressure level. Since the glass plane is transparent, a camera is setup at the bottom of the plane to capture the shape of the varying brush footprint. As both events are synchronized, the system would accurately capture the change of footprint caused by the brush deformation on glass plane.

However, we discovered that the previous brush footprint database acquisition method may not applicable to the recent 5-DOF robot. The deformation of a brush is critically determined by both the pressure exerted by the robot and the friction from the contacting surface. It is undoubted that the former factor is stably

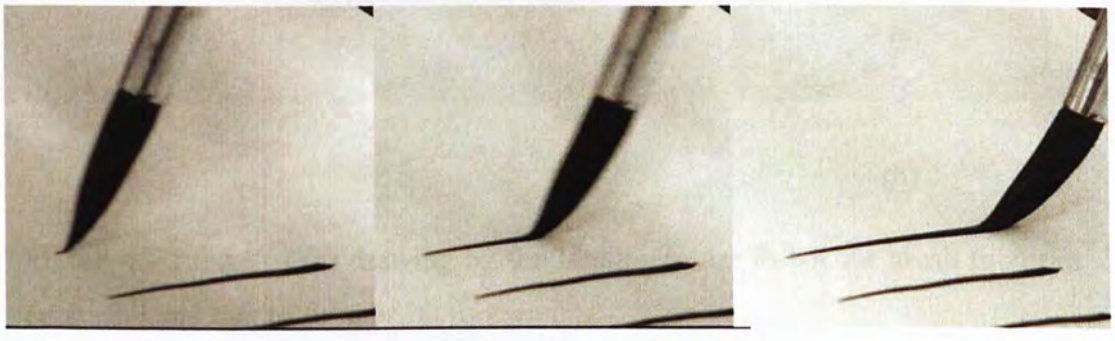
influenced by the motion of the manipulator. However, the friction to the brush is changed, if alternative materials are used to be the horizontal plane. So, to control the brush deformation, paper should be used to be the horizontal plane. Moreover, this model is formed by image database. It is discrete to the change of parameters, like inclination and pressure level. The resolution of such model mainly depends on the number of captured footprint images, which limits the adaptation for motion planning of the art robot. Also, the image database records the whole footprints. When it is applied to search trajectories in a Chinese calligraphy or painting, the entire shape of a footprint is mapped on a stroke. It would reduce the efficiency of the trajectory search process.

5.1. BRUSH FOOTPRINT EXTRACTION

Brush footprint is, basically, a water-droplet-like shape with a sharp tip. The proposed brush footprint acquisition method is, in principle, to approximate the shape. Hence, the shape boundary can be represented by several sample points. The positions of the points are varied when the pressure level and brush inclination are changed. Therefore, it is considered to be a mathematical model with several continuous functions obtained using curve fitting technique. Since the boundary is sampled, this approach would speed up the computational time for stroke path searching. Besides, as mentioned, the footprint image database does not contain continuous information of the brush footprint. The proposed model interpolates all the sampled points to entirely estimate the shape. To examine this method, the brush is constrained moving opposite to the direction pointed by the brush tip (opposite brush tip direction) to simply the problem.

When the brush is first pressed on a surface, the shape of the footprint is affected by the hairy brush. Some of the hairs are bent for a short period until the brush moves to refine them. Therefore, the model should predict the shape of footprint during motion, instead of the one in non-steady state. Figure 44 shows the

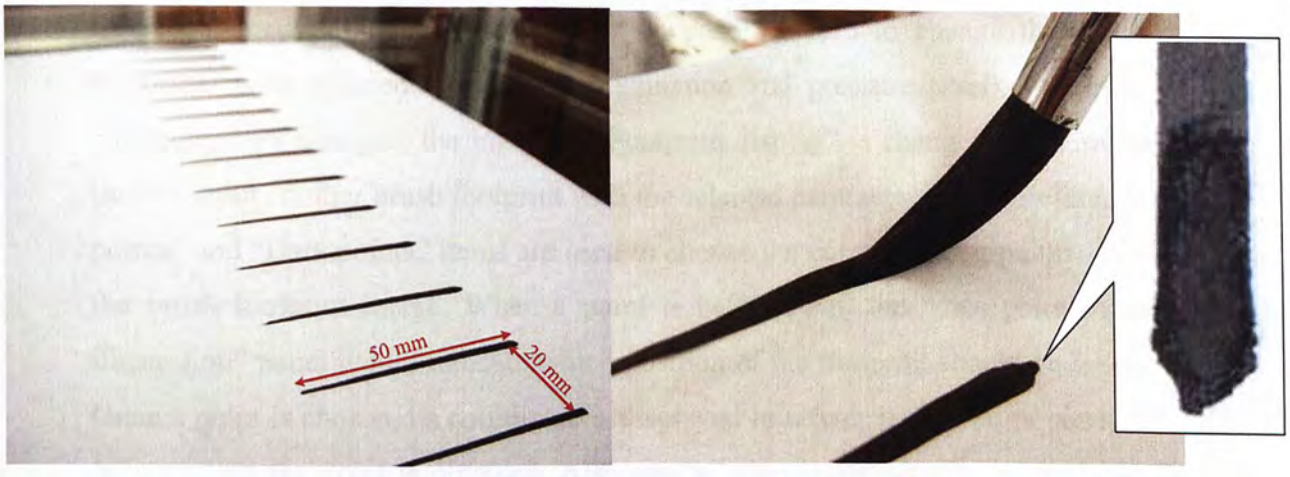
methodology to ensure the brush is in steady state (constant shape). The brush is first moved on the paper and adjusted to the required inclination, such that the brush tip is touching the on it (Figure 44(a)). Then, the manipulator move in the opposite brush tip direction and press it down to increase the pressure as shown in Figure 44(b). The brush further move in this direction until it locates in the desired position (Figure 44(c)).



(a) (b) (c)

Figure 44 (a)-(c). Images of brush motion to draw a stroke

The brush is stopped at the position for several seconds after drawing a stroke. This would condense extra ink in the footprint. The deeper color, so the footprint would be visualized. This is shown in Figure 45(b). From the stroke in the dialog, a footprint is display with darker color. The boundary can be apparently observed. The robot continues to draw the strokes with different inclination and applied pressure. Each stroke is 50 mm length and separated 20 mm interval (Figure 45). When all the strokes are drawn, they are converted into digital image using optical scanner. The footprints are sampled with sampled inclination from 5° to 50° every 5° and with sampled pressure (brush depth) from 1mm to 15mm every 1mm.



(a) (b)

Figure 45. Image of (a) drawing by the Intelligent Art Robot for brush footprint capturing and that of (b) a brush footprint

5.2. Graphical Interface for Inputting Sample Points of Brush Footprints

After capturing the brush footprint, a graphical user interface (GUI), which is shown in Figure 46, is used to input the data points of the footprints. Although the brush footprints can be observed to have a boundary separated from the background, it may not be easy to extract the foreground (the footprint) automatically by thresholding or other boundary extraction technique as the color of the footprint is determined by the optical input device and the density of the ink. The GUI provides an essential manner to manually input the specific points in a footprint.

In the GUI, the calibration panel controls the unit (i.e. mm/pixels) of an image. To make sure the unit is accurately input, an image containing a straight line with constant length is opened in the GUI. Both terminals of the line are selected by users. With the information, the GUI would calculate the required unit for input

images. Moreover, the parameter selection panel is used to change the brush footprints with different parameters (inclination and pressure level). Once the parameters are changed, the image in "Footprint display" is changed at the same time to input another brush footprint with the selected parameters. The "Reference points" and "Data points" items are used to choose the corresponding points from the brush footprint image. When a point is being input, the "data point input illustration" panel would indicate which position of the footprint should be taken. Once a point is chosen, its coordinate is displayed in reference/data point panel to indicate that the point is obtained. Also, an "o" sign is also drawn on the brush footprint display at the position of the data point. Therefore, users can make should that data points are accurately input.

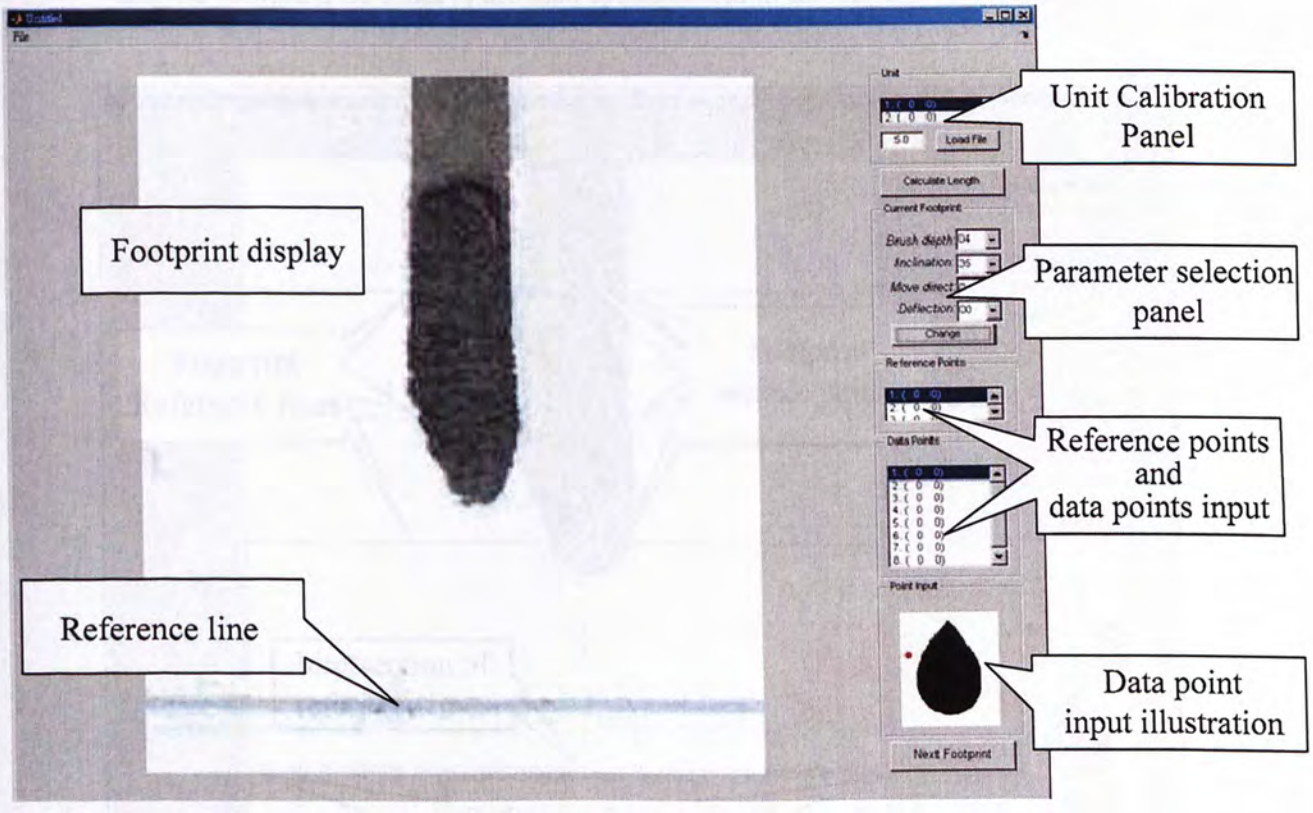


Figure 46. The GUI for footprint sample points input

When the sample points of a footprint is input, the reference points and the sample points in the ends of the brush footprints are selected. Two conceptual lines

(indicated in red color in Figure 47) are generated to denote the center line and the reference line of the footprint. Since the brush is deformed when the robot is writing on a surface, the coordinate origin of the brush footprint would not be located at the end of the footprint. It can be determined by the intersection of both (red dot) lines. Therefore, the coordinate of the data points are the corresponding position from this origin. Then, three footprint reference lines are drawing on the footprint image to indicate which points in the footprint boundary should also be input. As the vertical (red) reference line passes through the data points at both ends of the footprint, this reference line is approximately perpendicular. Also, the blue reference lines are parallel to the horizontal reference line. Consider a symmetric brush footprint, the data points (intersection of the blue reference lines and the footprint boundary) are also symmetrical to the vertical reference line.

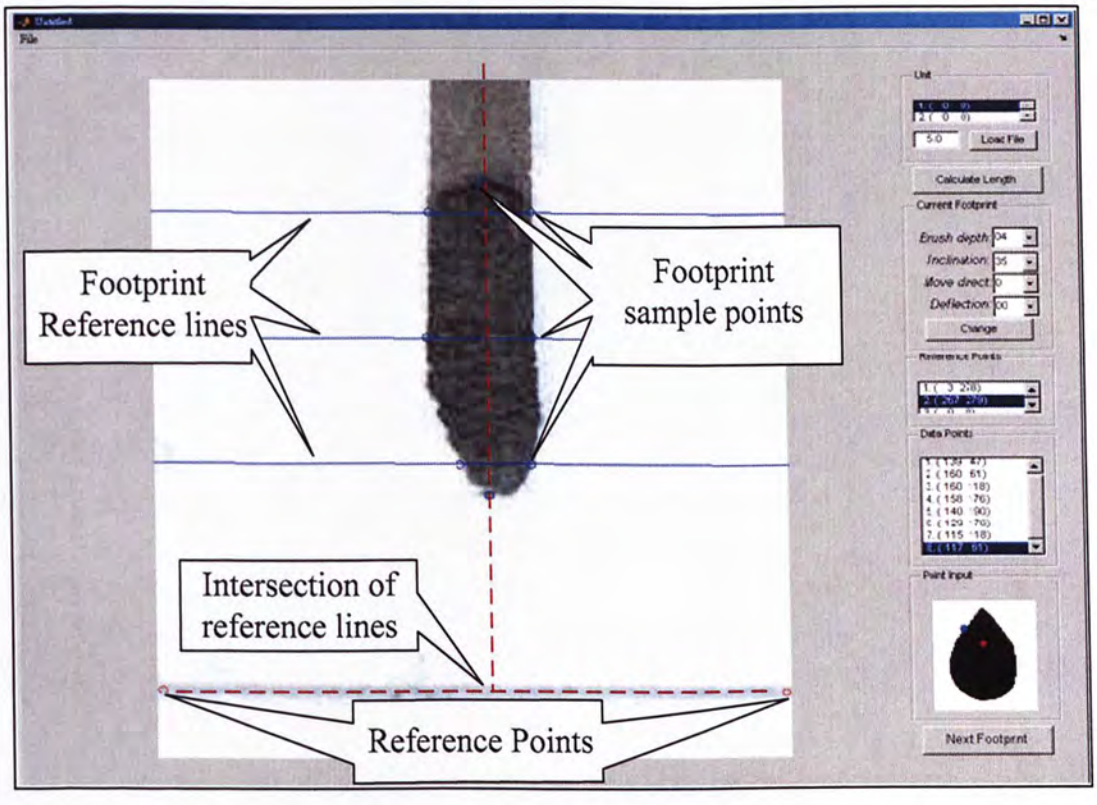


Figure 47. Illustration of reference lines after input the reference points and the sample points in the two ends of a footprint

The sampled brush footprints are shown in Figure 48 and Figure 49. Each sampled footprint is illustrated with different colors to indicate the separated footprints. Figure 48 shows the footprints varies in pressure level with constant brush inclinations. It can be observed that the sizes of the footprints are expended, when the brush pressure is increased. For the footprint shown in Figure 49, it demonstrates the relationship between the shapes of footprints to the brush inclination. The position of the footprint is moving from positive direction to negative direction, if the inclination is increasing. In this observation, the footprints show a tendency of variation when the parameters are changed. Therefore, it also shows that curve fitting technique would potentially approximate this tendency.

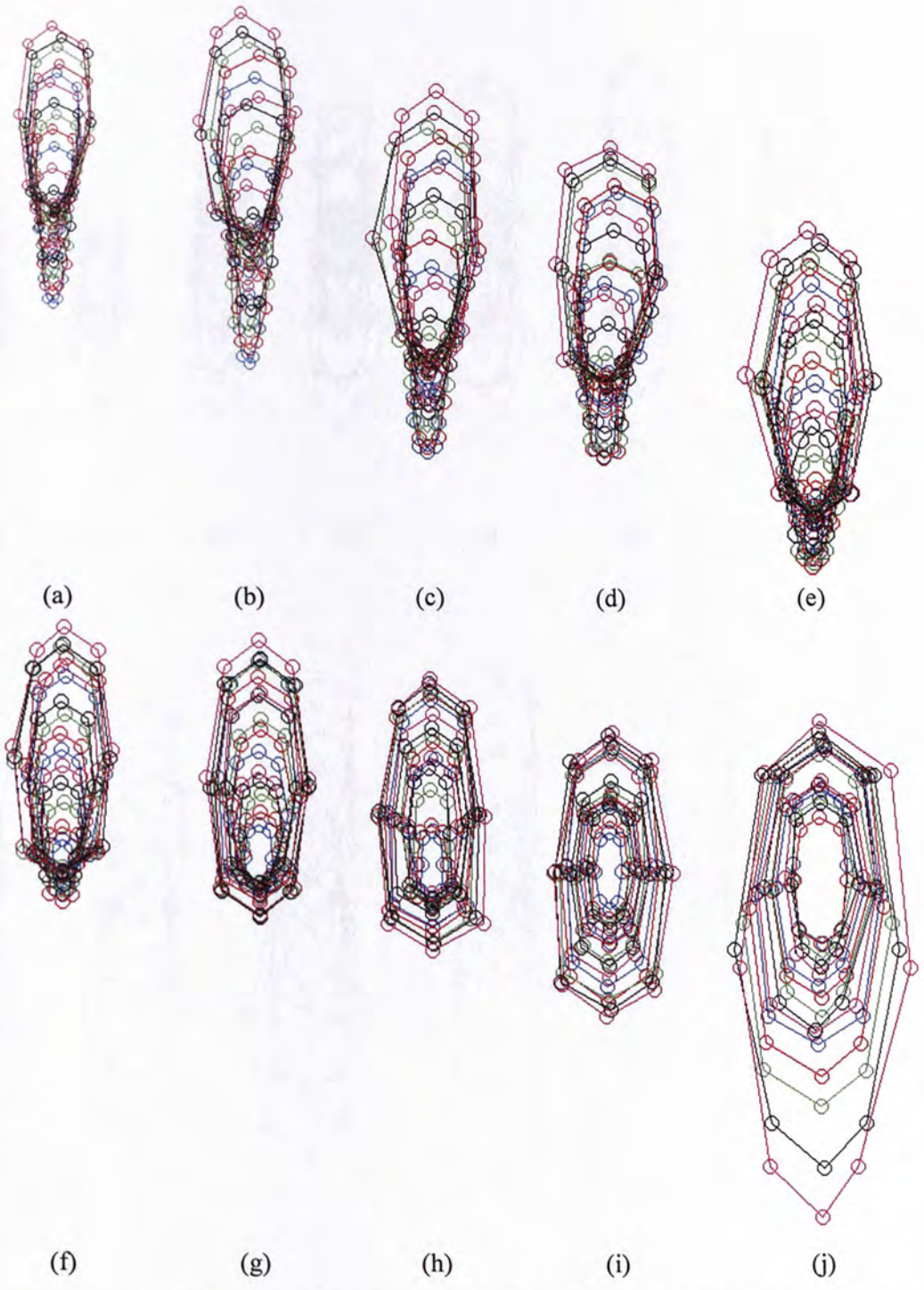


Figure 48(a)-(j). The sample points of the extracted footprints with change in brush pressure level (1mm-15mm)

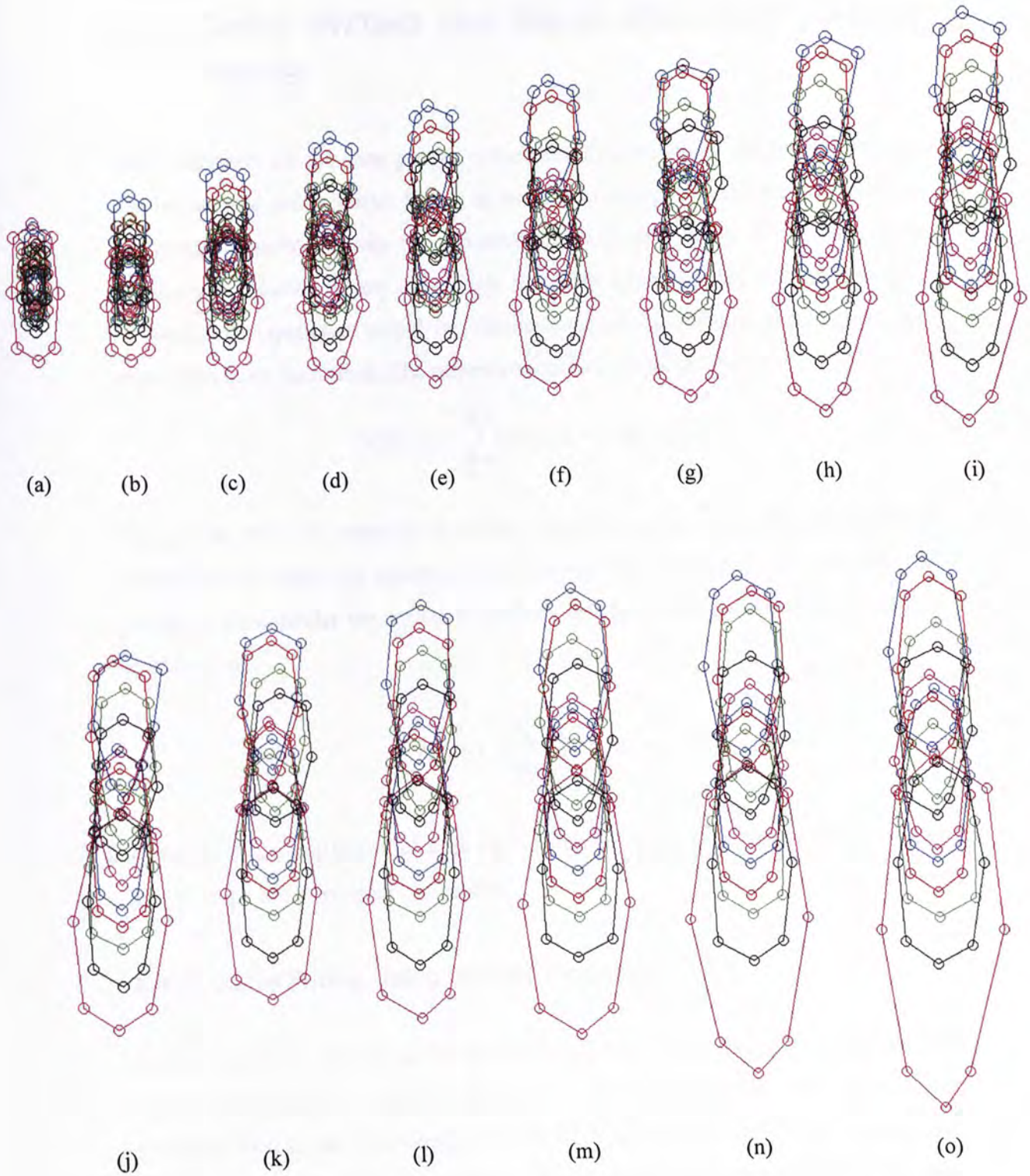


Figure 49(a)-(o). The sample points of the extracted footprints with variations in brush inclination from (5°-50° with 5° increment)

5.3. CURVE FITTING FOR BRUSH FOOTPRINT SAMPLE POINTS

After sampling all the data points, a mathematical model of the brush footprint can be derived using curve fitting technique to interpolate the data points into continuous functions with two parameters, including brush depth and brush vertically inclination. Every coordinate should be approximated with an equation, so totally 16 equations would be obtained. Genetic algorithm and least square regression have been tried. The objective function of the problem is

$$V(d,\gamma) = \sum_{t=1}^{n} (x(d,\gamma) - y(d,\gamma))^2$$

where $V(d,\gamma)$ is the objective function, $y(d,\gamma)$ is a data point. $V(d,\gamma)$ should be minimized to obtain an equation that close to the data points. To simplify the problem, we consider the equation obtained by curve fitting to be 6^{th} order, and it is in form of

$$x(d,\gamma) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij} d^i \gamma^j$$

where a_{ij} is the coefficient of term i-j, d is the brush depth inserting into the paper and γ^j is the brush vertical inclination.

5.3.1. Curve Fitting Using Genetic Algorithm

In this problem, the objective of the GA-based curve fitting is to search the optimal values of a_{ij}, such that the cost function is minimized. The chromosome structure used in the GA-based curve fitting is shown in Figure 50. P_i denotes the i^{th} parameter in the 6^{th} order equation. Each parameter is represented by a 16-bit floating point number.

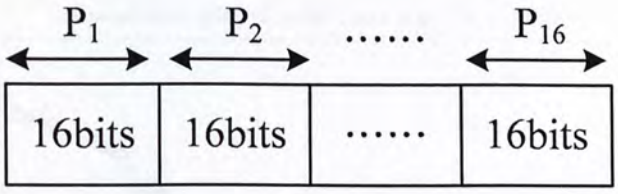


Figure 50. Chromosome structure with 16-bit floating point representation in 16 coefficients.

The GA-based curve fitting was performed on a PC with Pentium D 3.2GHz CPU and 2GB RAM. The program was running using matlab in Windows XP platform. The parameters for the GA are shown in Table 2. The initial population size is 100 chromosomes. Every chromosome is represented by 256-bit floating point representation with 16 parameters. And, the searching range of the coefficients is within 1 and -1. In the main program, the maximum number of generation is set to be 30000. The crossover and mutation rate are 0.7 and 0.001 respectively.

.GA parameter	Value
Population size	100
Chromosome length	256
Searching range of coefficients	1 to -1
No. of generation	30000
Crossover rate	0.7
Mutation rate	0.001
No. of combinations in the chromosome	$2^{256}=1.158 \times 10^{77}$

Table 2. Summary of the parameters for GA-based curve fitting

A result for the x-coordinate of point 1 is shown in Figure 51. The computational time is 27 hours 38.2 mins. The original fitness is around 2800. Although the result converges to fitness of around 500, it cannot be minimized close to zero.

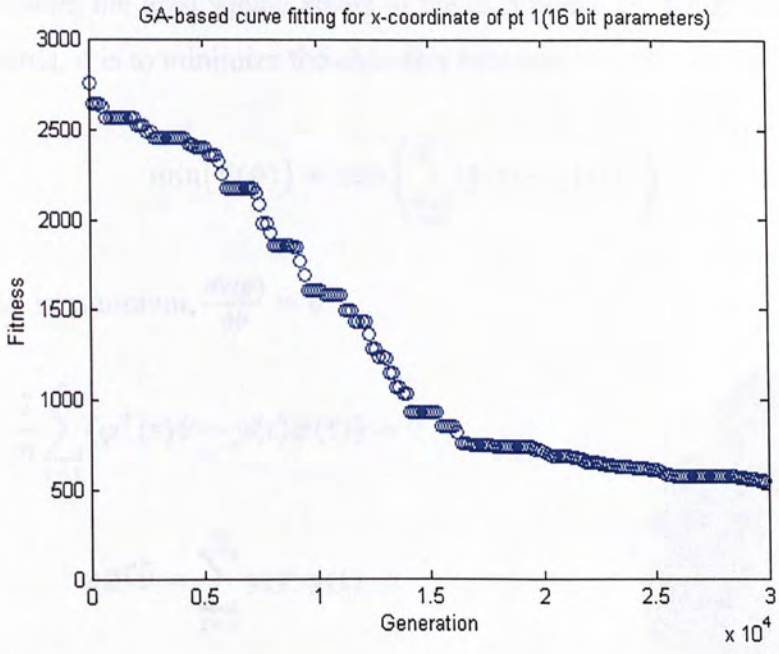


Figure 51. The fitness of GA-based curve fitting in each generation

5.3.2. Curve Fitting by Least Squares Regression

According to the fitness of the GA-based approach, it is failed to find equations to fit the data points. Therefore, least squares regression [50] is used improve the problem. Consider the equation obtained by curve fitting to be:

$$y(t) = \sum_{i=1}^{m} (b_i u_i)$$

where $u_i = z^j \gamma^k$ (z is the pressure level and γ is the brush inclination)

Let $\theta = [b_1 \quad \dots \quad b_m]^T$ and $\varphi = [u_1 \quad \dots \quad u_m]^T$,

Therefore, $$y(t) = \varphi^T \theta$$

Let $\tilde{\theta}$ be the estimated parameters and $\tilde{y}(t) = \varphi^T \tilde{\theta}$ be the predict output

In this problem, the least square errors of the data points are being minimized. In another words, it is to minimize the objective function:

$$\min(V(\theta)) = \min\left(\sum_{t=1}^{n}(\tilde{y}(t) - y(t))^2\right)$$

When $V(\theta)$ is minimum, $\frac{\partial V(\theta)}{\partial \theta} = 0$

$$\frac{2}{n}\sum_{t=1}^{n}(\varphi^T(t)\theta - y(t)\varphi(t)) = 0$$

$$\emptyset\emptyset^T\tilde{\theta} = \sum_{t=1}^{n} y(t)\varphi(t)$$

Where $\emptyset = [\varphi(1) \quad \ldots \quad \varphi(n)]$ for $m \leq n$

$$\tilde{\theta} = (\emptyset\emptyset^T)^{-1}\left(\sum_{t=1}^{n} y(t)\varphi(t)\right)$$

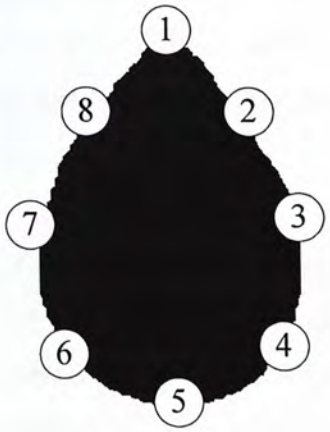


Figure 52. Indication of sample points

After solving the equation above using the data points of brush footprints, 16 equations were obtained. The equations and the data points are compared. The sample point number is indicated in Figure 52. The maximum errors between them are shown in Table 3. The errors are small, which implies that the equations can be used as mathematical model for the brush footprint. In point 1 and point 5, the errors of y-coordinates are relatively large. However, it is observed that the large error is caused when brush depth is at the maximum of the range, i.e. 15mm. So, the range for brush depth is reduced to 0 to 14mm to prevent excessive error in the trajectory for IRAS.

	Point 1	*Point 2*	*Point 3*	*Point 4*	*Point 5*	*Point 6*	*Point 7*	*Point 8*
x	1.1152	1.2467	1.151	1.5594	1.657	1.4953	1.6713	1.6347
y	2.4187	1.3353	1.4956	1.9578	3.484	1.8483	1.4353	1.4490

Table 3. Maximum error compared with the sampled points (mm)

5.4. DISCUSSION

When GA was applied to curve-fit the sample points of brush footprints, the cost function did not converge to zero, and become stable at a high score. The reason for the failure is that the number of combinations in the chromosome is too large to search the global optimal solution. The total combination number is 1.158×10^{77}. Although 30000 iterations are processed, only 2.1×10^{6} combinations are explored. The ratio of the searched combination tends to zero. Therefore, GA is not suitable to obtain the coefficients of the equations.

For the equations obtained from least square regression, they are low order (only 6-order) equations. So, the equations have limited numbers of maxima and minima. For this reason, the values between peak values are either strictly increasing or strictly increasing, local optimization technique, like steepest method, may be applied to find the desired values in trajectories for IRAS, if the range of the value is chosen. This would reduce the computational time for searching, as global optimization is a time consuming process.

Moreover, the brush footprint model was obtained by writing strokes on same type of papers to ensure constant ink diffusion. If other types of papers are used, the diffusion rate would be different. This may vary the amount of ink condenses on the brush footprint, so it would affect the spreading and formation of the shape. To avoid such problem, the trajectory execution speed in IRAS can be adjusted. If a paper has higher diffusion, the brush should move faster. For the one with lower diffusion rate being used, the execution speed should be reduced.

Chapter 6: TRAJECTORY GENERATION FOR ROBOTIC CHINESE CALLIGRAPHY

The basic idea of calligraphic trajectory generation is to map brush footprints to fully cover the strokes in a Chinese word. A method based on GA is proposed in [51] and [52] to create 3-DOF trajectories. It superimposes brush footprints along the medial axis of strokes. The sizes of the footprints are changed, depends on the random strategy in GA, both the outside area of footprints and uncovered regions are minimized. However, since GA is used, the time required for obtaining a trajectory would be long, especially in complicated problems, e.g. 5-DOF trajectory searching. Therefore, in this chapter, a method for fast calligraphic trajectory generation is proposed to reduce the computational time.

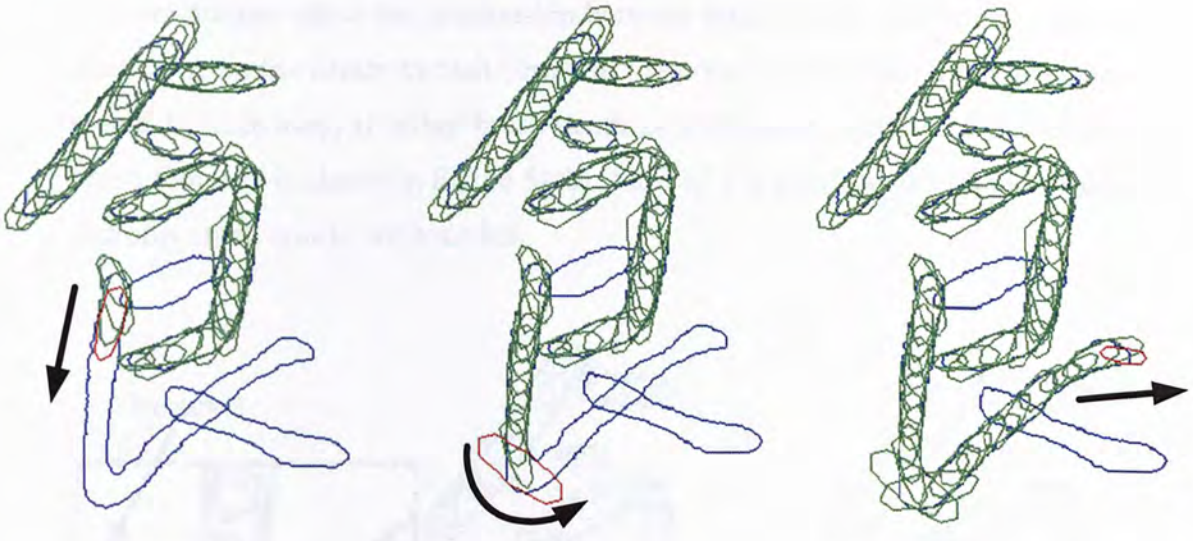
6.1. STROKE TRAJECTORY SEARCHING WITH ACCORDING STROKE WIDTH

To simplify this problem, some conditions should be fulfilled to limit the searching range of trajectory parameters. In acquisition of brush representation, discussed in Chapter 5, the brush manipulator is moving along the longitudinal direction (direction from point 1 to point 5) of brush footprint to induce residual brush footprints. Hence, when the brush is used to draw with the same parametric values in brush depth, vertically inclination and direction of motion, it should imprint a brush footprint with the same shape. If the brush moves in the other directions, it would change the friction between the brush and the paper. The footprint would be deformed differently comparing to the one with constant parameters. Therefore, to reduce the possibility of variation, the direction of movement is limited, which is an important criterion, in this calligraphic trajectory searching.

Moreover, the shape of brush footprint is symmetrical along its longitudinal axis. It has the largest width in the middle part, which is close to the distance between point 3 and point 7. Hence, the distance is $X_3(d,\gamma) - X_7(d,\gamma)$, where d is the brush depth, γ is the brush vertical inclination. The mid-point of the points is known as center of a brush footprint. Besides, the refined internal edges in a stroke are cross-section lines, which can also be considered as the stroke widths. So, the center of brush footprint is supposed to touch the medial axis. The objective function F_{obj4} to minimize the difference of stroke width and the width of brush footprint is

$$F_{obj4}(d,\gamma) = X_3(d,\gamma) - X_7(d,\gamma) - W_{stroke} \quad (4)$$

where $X_3(d,\gamma)$ and $X_7(d,\gamma)$ are the x coordinates of point 3 and point 7 in the sample points of brush footprint, and W_{stroke} is the stroke width. The starting point of a trajectory can be initialize with constant value (d,γ), say (2,20°). For the other points, as mentioned in Chapter 5 since the brush footprint representation is expressed by several 6-order equations, steepest method is applied to find the values that minimizes F_{obj4}. The trajectory searching process is as illustrated in Figure 52. The red shape indicates the brush footprint that is mapping on the stroke. Even the medial axis turns (Figure 53(b)), the longitudinal axis also follows the direction of it. This mapping continues, until it reaches to the other end of the stroke (Figure 53(c)).



(a) (b) (c)

Figure 53. Trajectory searching along the medial axis.

6.2. IMPROVEMENT IN STROKE TRAJECTORY

After mapping, a trajectory can be obtained by interpolating the positions of the brush footprint centers. However, if only the width is considered, some of the footprints may superimpose a stroke outside its boundary as shown in Figure 54(a). The distance, ΔD, between vertex 1 and the stroke boundary is largest among the vertices in the same footprint, i.e.

$$\Delta D = \sqrt{(X - X_i)^2 + (Y - Y_i)^2}$$

Therefore, the size of this outlier should be reduced, such that it just touches the boundary. At the same time the change in brush depth and vertically inclination should also be reduce to prevent vibration in the robot. Therefore, the objective function can be written as

$$F_{obj5} = \omega_1 \Delta D + \omega_2 \Delta I + \omega_1 \Delta B \quad (5)$$

where ΔD is the distance of vertex 1 and the closest point in the boundary, ΔI and ΔB are the change in the brush inclination and depth respectively. Figure 55(a)-(f)

plot the meshes show the relationship between brush depth, inclination and the distance from the center to each vertex in footprint. It shows that these distances would be increased, if either brush depth or inclination increases. The resized brush footprint is shown in Figure 54(b). Most of it is overlapped with the stroke, and only small area is not bounded.

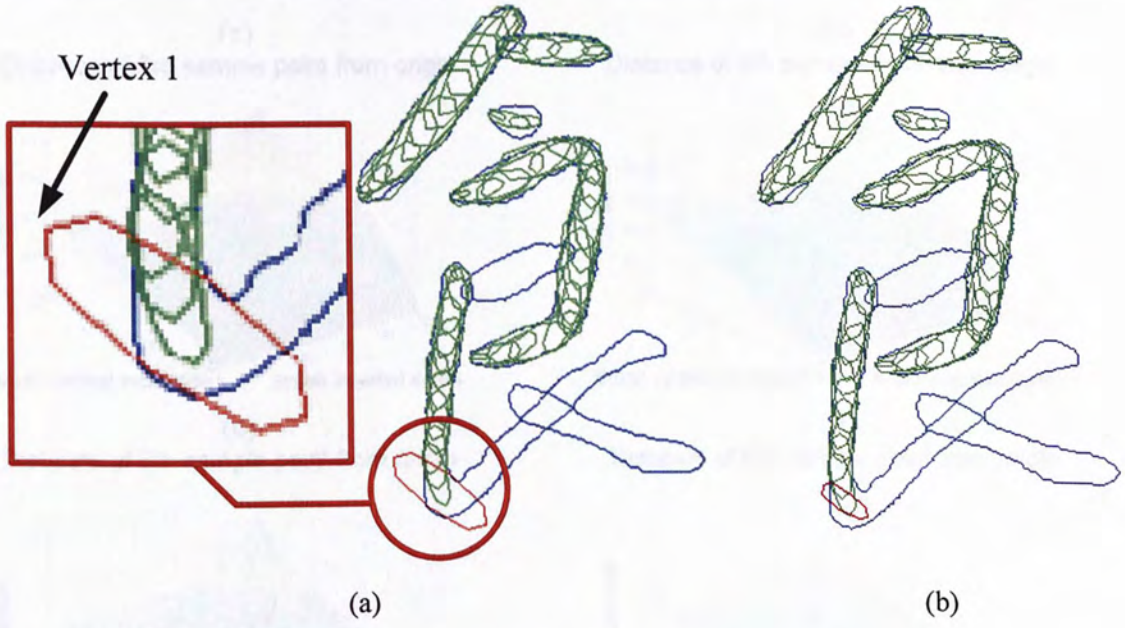


Figure 54. Improvement in stroke trajectory.

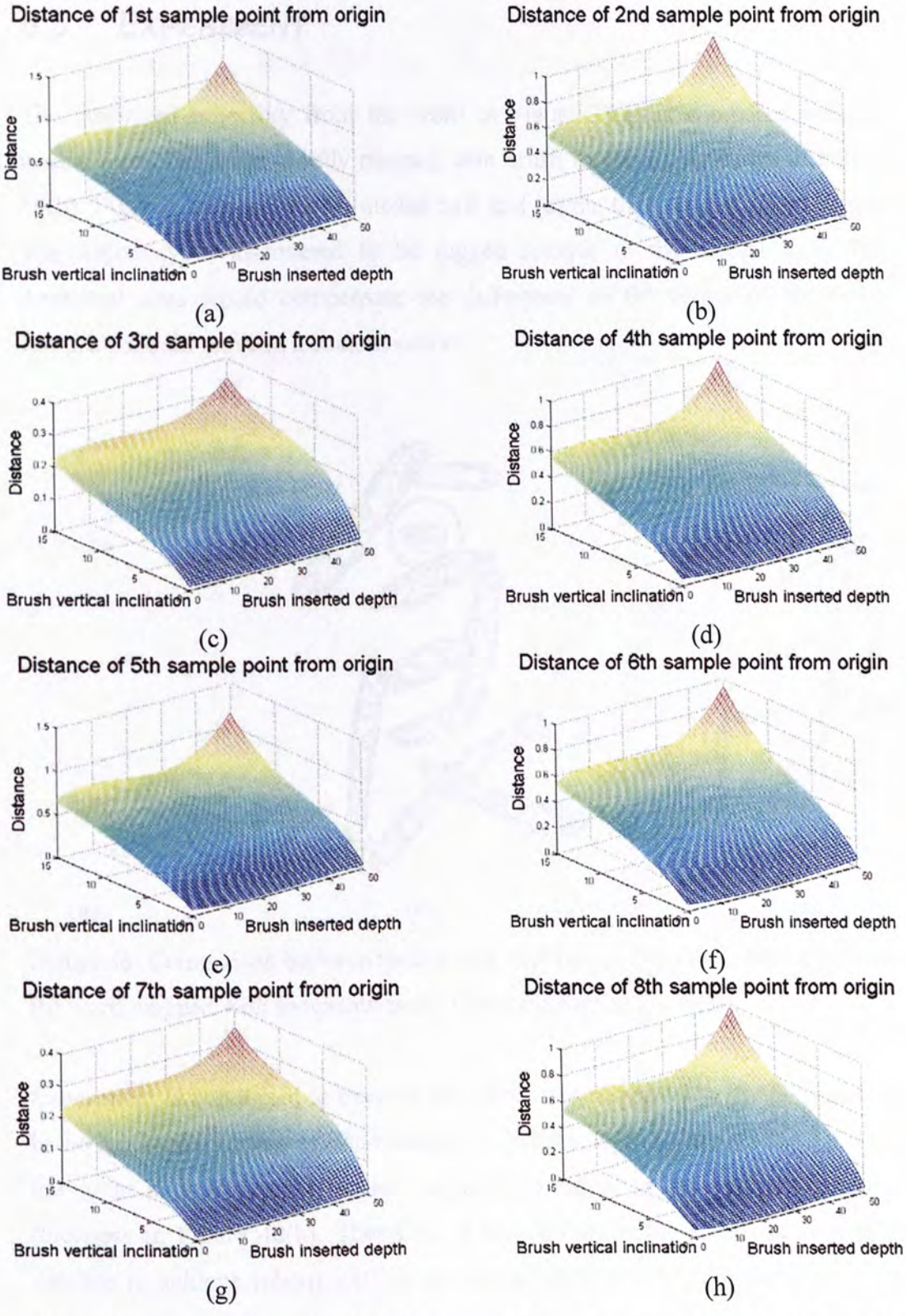
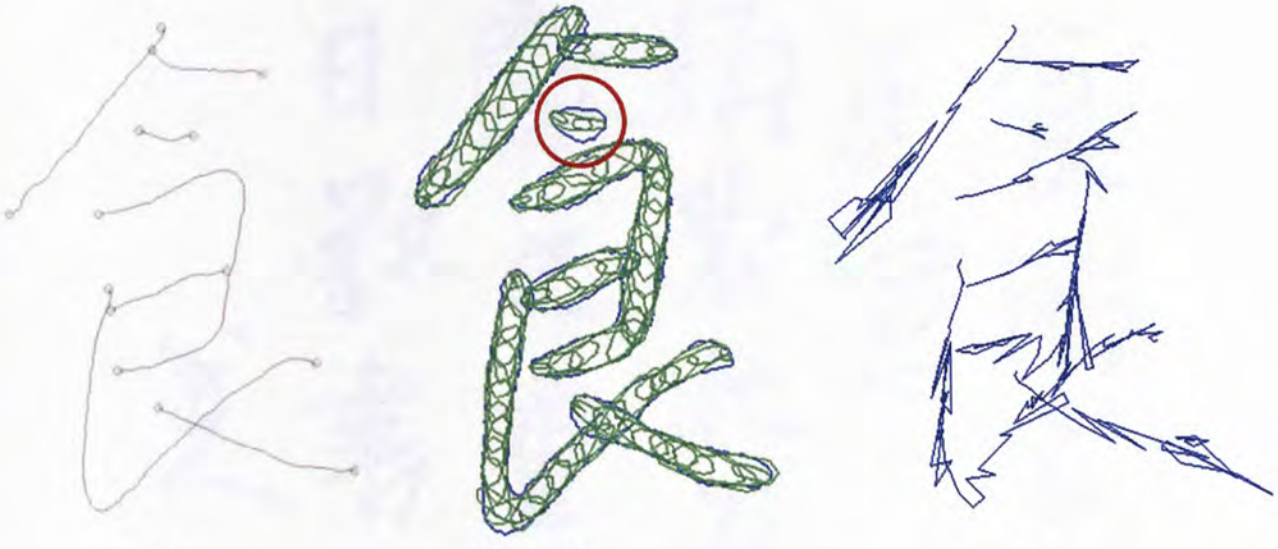


Figure 55. (a)-(h) Plots show the distances of sample points from brush footprint origin

6.3. EXPERIMENT

The generated trajectory from the word in Figure 38(a) is compared with its medial axis. The word is fully mapped with brush footprints as shown in Figure 56(b). Figure 56 shows the (a) medial axis and (c) the trajectory of x and y axis. The trajectory is discovered to be rugged relative to the medial axis. The rotational axes would compensate the difference, so the center of the brush footprint should move in a smooth curve.



(a) (b) (c)

Figure 56. Comparison between medial axis and the x-y trajectory: (a) medial axis, (b) word mapped with footprints and (c) the trajectory of x-y axes

Experiment is conducted to execute the 5-DOF trajectory using IRAS. Figure 57 is the replicated version of Bada Shanren's calligraphy in Figure 40(a). The result has great improvement in stroke thickness compared to the one with constant thickness in Figure 40(b). Therefore, it shows that the trajectory generation is capable to achieve robotic calligraphy using hairy brush. However, one of the strokes does not fully covered by footprints, which is indicated by a red circle in Figure 56(b). The uncovered regions are caused by footprints, which do not accent

with the stroke width, in the middle part of the stroke. Therefore, in calligraphy shown in Figure 57, the stroke in the corresponding word would be thinner than the desired stroke width. This problem is different to be solved, as the brush footprint model does not provide shapes that both superimpose the stroke with no uncovered region and have width some as the corresponding cross-section lines. To alleviate the problem, the tolerance between stroke boundary and the feature points in the footprint should be increased, so brush footprints with thicker width would be used.

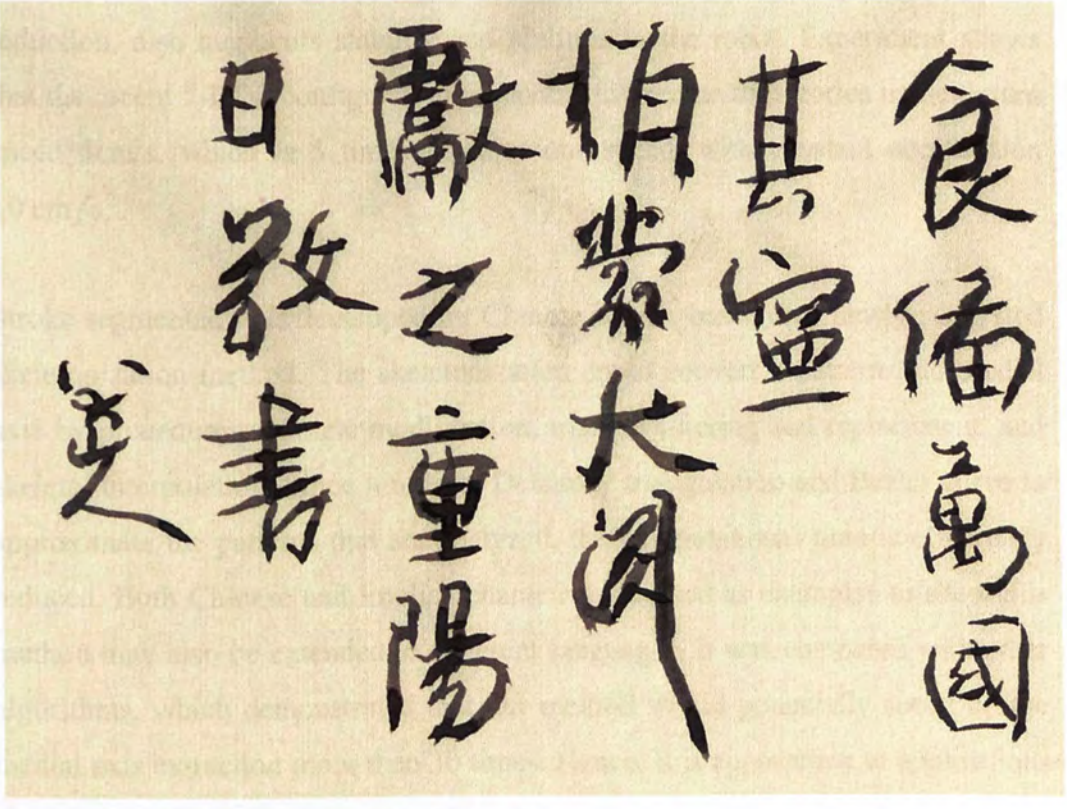


Figure 57. Replication of in Bada Shanren's calligraphy using IRAS

CONCLUSION AND FUTURE WORK

In this thesis, Robotic Chinese calligraphy is discussed, which achieved by a robot artist IRAS using hairy brush. Stroke segmentation and trajectory generation are the major algorithms to fulfill the task in our approach. Moreover, to satisfy the requirements for writing Chinese calligraphy, IRAS has been improved in hardware for dexterous motion. The modification includes additional rotational DOFs in manipulator and axial-rotary brush to enable the brush manipulator inclines. Some auxiliary functions, like manipulator positioning and vibration reduction, also augments stability and abilities to the robot. Experiment shows that the recent 5-DOF configuration is capable to execute trajectories in maximum speed 5cm/s, which is 5 times the previous speed, with constant acceleration $10\ \mathrm{cm/s^2}$.

Stroke segmentation is developed for Chinese words, based on a newly suggested skeletonization method. The skeletonization could convert a pattern into medial axis by procedures: triangle modification, triangle filtering and replacement, and skeletal interpolation. Since it utilizes Delaunay triangulation and Bezier curve to approximate the patterns that are analyzed, the computational time is effectively reduced. Both Chinese and English characters are used as examples to show this method may also be extended to different languages. It was compared with other algorithms, which demonstrated that our method would potentially speed up the medial axis extraction more than 30 times. Hence, it is appropriate to applications that high resolution images are used.

Medial axis is essential information to determine the connectivity of the strokes in a word. To convert it into a continuous stroke, the triangular mesh is adjusted based on the structure of Chinese words. Spurious branch removal filters the unrelated branches and branch matching determines the branch correspondences. Bezier curve is also applied to refill the vanished regions. Experiment is conducted using calligraphy of one of the most famous ancient Chinese

calligrapher, Bada Shanren. It shows that this approach is feasible to extract most of the strokes in the Chinese words. However, some of the strokes cannot be extracted correctly. The problem is caused by the dominant of some of the factors. Therefore, fuzzy logic is proposed to improve weighting the branch matching.

Furthermore, brush footprint representation is acquired from experiment, which samples the footprint boundary. Both Least square regression and GA are applied to interpolate the boundary sample points. Result shows that, if ranges of parameters are selected, it would generate brush footprints with small error (less than 2mm difference to the sample points). It can be used as a model to predict the shape of brush footprint varied by brush deformation. However, the brush footprint would also be varied by the current status of the brush. So, to improve the brush footprint representation, brush deflection and direction of motion should also be considered.

After obtaining the segmented strokes and footprint representation, trajectory is searched along the stroke medial axes. In essence, it is generated by mapped brush footprints to the strokes, such that the footprints just touch the stroke boundary. The mechanical output executed by IRAS is compared with the one with constant stroke width and the original calligraphy. Result shows that the replicated calligraphy with 5-DOF trajectory is superior, since its stroke width is closer to the original version. Apart from Bada Shanren, calligraphies written by other most famous artists, e.g. Wang Xizhi and Yan Zhenqing, will also be replicated using the proposed procedures in future works. Success of the work would potentially realize robotic Chinese calligraphy.

REFERENCES

[1] F. Hara., "Artificial Emotion of Face Robot through Learning in Communicative Interactions with Human", Proceedings of the 13th IEEE International Workshop on Robot and Human Interactive Communication, pp. 7 - 15, 2004.

[2] T. Fukuda, J. Taguri, F. Arai, M. Nakashima, D. Tachibana, Y. Hasegawa, "Facial expression of robot face for human-robot mutual communication", Proceedings of International Conference on Robotics and Automation (ICRA '02), vol 1, pp. 46-51, 2002.

[3] R. Dance, K. Kosuge, T. Hayashi, Y. Hirata and R. Tobiyama, "Dance Partner Robot – Ms", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 4, pp. 3459 – 3464, 2003.

[4] J. Or and A. Takanishi, "A Biologically Inspired CPG-ZMP Control System for the Real-time Balance of a Single-legged Belly Dancing Robot," Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 931 – 936, 2004.

[5] H. Morgan, Harold Cohen's Aaron, "The Robot as an Artist". http://www.scinetphotos.com/aaron.html

[6] P. McCorduck, "Aaron's Code: Meta-art, Artificial Intelligence and the work of Harold Cohen", W. H. Freeman & Co., New York, New York Copyright 1991.

[7] http://artbots.org/2003/participants/Drawing_Machine/

[8] http://zanelle.wordpress.com/

[9] L. Pagliarini, and H.H. Lund, "Art, Robots, and Evolution as a Tool for Creativity", Creative Evolutionary Systems, 2001.

[10] A. Srikaew, M.E. Cambron, S. Northrup, R.A. Peters II, D.M. Wilkes, and K. Kawamura, "Humanoid Drawing Robot", IASTED International Conference on Robotics and Manufacturing, Banff, Canada, July, 1998.

[11] D. Morris, K. Phelps, and N. Joshi, The drawing teleoperated robot. http://techhouse.brown.edu/~neel/drawing_telerobot

[12] P. Monaghan, "An art professor uses artificial intelligence to create a computer that draws and paints", The Chronicle of Higher Education, pp. 27-28, May 1997.

[13] Y. Yam, K.W. Lo, K.W. Kwok, "A Robot Drawing System: Preliminary Design and Demonstration", Proceedings of the 2003 International Conference on Intelligent Technologies, Chiang Mai, Thailand, pp. 545-551, 17-19 Dec. 2003.

[14] Josh H. M. Lam, K. W. Lo, Y. Yam, "Robot Drawing Techniques for Contoured Surface Using an Automated Sketching Platform", Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering, Scottsdale, Arizona, USA, September, 2007.

[15] H. Fujiyoshi, A. J. Lipton, "Real-time human motion analysis by image skeletonization", Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV), pp. 15-21, Oct. 1998.

[16] R. M. Haralick and L. G. Shapiro, "Computer and Robot Vision," Reading, MA, Addision-Wesley, 1992.

[17] M. W. Wright, "Skeletonisation as model based feature detection," Proceedings of International Conference on Image Processing and its Applications, vol. 2, pp. 18-20, Aug. 1997.

[18] L. Lam , S. -W. Lee, and C. Y. Suen, "Thinning Methodologies – A Comprehensive Survey," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 14, pp. 869-885, Sept. 1992

[19] T.Y. Zhang, C.Y. Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns," Communication of the ACM, vol. 27, pp. 236 - 239, 1984

[20] M. Senthilanyaki, S. Veni, K. A. N. Kutty, "Hexagonal Pixel Grid Modeling for Edge Detection and Design of Cellular Architecture for Binary Image Skeletonization," Proceedings of Annual India Conference (INIDCON), pp. 1-6, Sept 2006.

[21] R. M. Ralenichka, M. B. Zaremba, "Multi-scale model-based skeletonization of object shapes using self-organizing maps", Proceedings of the 16th International Conference on Pattern Recognition, vol. 1, pp. 143-146, Aug. 2002.

[22] S. M. Pizer, W. R. Oliver and S. H. Bloomberg, "Hierarchical shape description via the multi-resolution symmetric axis transform," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 9, No. 4, pp.505-511, 1987.

[23] H. Rom and G. Medioni., "Hierarchical decomposition and axial shape description," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 15, pp. 973-981, 1993.

[24] J. J. Zou and H. Yan, "Skeletonization of ribbon-like shapes based on regularity and singularity analyses," IEEE Transaction on Systems, Man and Cybernetics, vol. 31, pp. 401-407, Jun. 2001

[25] Josh H. M. Lam, Y. Yam, "A Skeletonization Technique Based on Delaunay Triangulation and Piecewise Bezier Interpolation", Proceedings of the 6th IEEE International Conference on Information, Communications and Signal Processing, Singapore, December, 2007 (Accepted).

[26] Josh H. M. Lam, Y. Yam, "A Skeletonization Method Using Delaunay Triangulation for Ribbon-like Patterns", The Second Beijing-Hong Kong International Doctoral Forum 2007, Hong Kong, July, 2007

[27] P. J. Giblin and S. A. Brassett, "Local Symmetry of Plane Curves", The American Mathematical Monthly, vol. 92, no. 10, pp. 689-707, Dec. 1985.

[28] A. Rosenfeld, "Axial Representation of Shape", Computer Vision, Graphics and Image Processing, vol. 33, pp. 156-173. 1986

[29] M. J. Brady, H. Asada. "Smooth Local Symmetries and Their Implementations". International Journal of Robotics Research, vol. 3, no. 3, 1984.

[30] J. Ponce, "On Characterizing Ribbons and Finding Skewed Symmetries", Proceedings of International Conference on Computer Vision, Graphics, and Image Processing, vol. 52, pp. 328-340, 1990

[31] H. Zabrodsky, "Computational Aspects of Pattern Characterization – Continuous Symmetry". pp. 13 – 21, 1993

[32] H. S. Chang, and H. Yan, "Analysis of Stroke Structures of Handwritten Chinese Characters," IEEE Transaction on System, Man and Cybernetics, vol. 29, pp. 47-61, Feb. 1999.

[33] V. Pervouchine, G. Leedham, K. Melikhov, "Three-stage handwriting stroke extraction method with hidden loop recovery," Proceedings of the 8th International Conference on Document Analysis and Recongnition, vol. 1, pp. 307-311, Aug. 2005.

[34] D. T. Bui, "Classifying Online Handwriting Characters under Cosine Representation", Proceedings of the 6th International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007), pp. 206 – 211, 22-24 Aug. 2007.

[35] Yin Liu, Yajie Yu, Wenyin Liu, "Online segmentation of freehand stroke by dynamic programming", Proceedings of the 8th International Conference on Document Analysis and Recognition, vol. 1, pp 197 – 201, 29 Aug.-1 Sept. 2005.

[36] Y. K. Chen, J. F. Wang, "Segmentation of handwritten connected numeral string using background and foreground analysis", Proceedings of the 15th International Conference on Pattern Recognition, vol. 2, pp. 598 - 601, 3-7 Sept 2000.

[37] Y. Chen, G. Leedham, "Independent component analysis segmentation algorithm", Proceedings of the 8th International Conference on Document Analysis and Recognition, vol. 2, pp. 680 – 684, 29 Aug. - 1 Sept. 2005.

[38] J. Zeng, Z. Q. Liu, "Stroke Segmentation of Chinese Characters Using Markov Random Fields", Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006), vol. 1, pp. 868 – 871, 2006.

[39] F. Chang, Y. C. Chen, H. S. Don, W. L. Hsu, C. I. Kao, "Stroke segmentation as a basis for structural matching of Chinese characters", Proceedings of the 2nd International Conference on Document Analysis and Recognition, pp. 35 – 40, 20-22 Oct. 1993.

[40] T. Ban, C. S. Zhang, W. Shu, Z. B. Kou, "A novel approach in off-line handwritten Chinese character stroke segmentation", Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2003), pp. 314 – 318, 27-30 Sept. 2003

[41] H. H. Chang, H. Yan, "Analysis of stroke structures of handwritten Chinese characters", IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 29, issue 1, pp. 47 – 61, Feb. 1999.

[42] Y. Tang, R. Chen, Y. H. Qiu, "An effective stroke extraction model for Chinese characters", Proceedings of 2003 International Conference on Machine Learning and Cybernetics, vol. 5, pp. 3193 – 3198, 2-5 Nov. 2003.

[43] K. Liu, Y. S. Huang, C. Y. Suen, "Robust stroke segmentation method for handwritten Chinese character recognition", Proceedings of the 4th International Conference on Document Analysis and Recognition, vol. 1, pp. 211 – 215, 18-20 Aug. 1997.

[44] J. W. Tai, Y. J. Liu, L. Q. Zhang, "A model based detecting approach for feature extraction of off-line handwritten Chinese character recognition", Proceedings of the 2nd International Conference on Document Analysis and Recognition, pp. 826 – 829, 20-22 Oct. 1993.

[45] Horace H. S. Ip, Helena T. F. Wong, "Calligraphic Character Synthesis using Brush Model," Computer Graphics International 1997 (CGI'97), pp. 13-21, Jun. 1997

[46] Nelson S. H. Chu and C. L. Tai, "Real-time Painting with an Expressive Virtual Chinese Brush", IEEE Computer Graphics and Applications, vol. 24, issue 5, pp. 76-85, Sept. 2004.

[47] Fenghui Yao; Guifeng Shao; Painting brush control techniques in Chinese painting robot, IEEE International Workshop on Robot and Human Interactive Communication, Roman, pp. 462 – 467, 13-15 Aug. 2005.

[48] N. S. H. Chu, C. L. Tai, "Real-time painting with an expressive virtual Chinese brush", IEEE Computer Graphics and Applications, vol. 24, issue 5, pp. 76 – 85, Sept. 2004.

[49] X. Mi, J. Xu, M. Tang, J.X. Dong, "The droplet virtual brush for Chinese calligraphic character modeling", Proceedings of the 6th IEEE Workshop on Applications of Computer Vision (WACV 2002), pp. 330 – 334, 3-4 Dec. 2002.

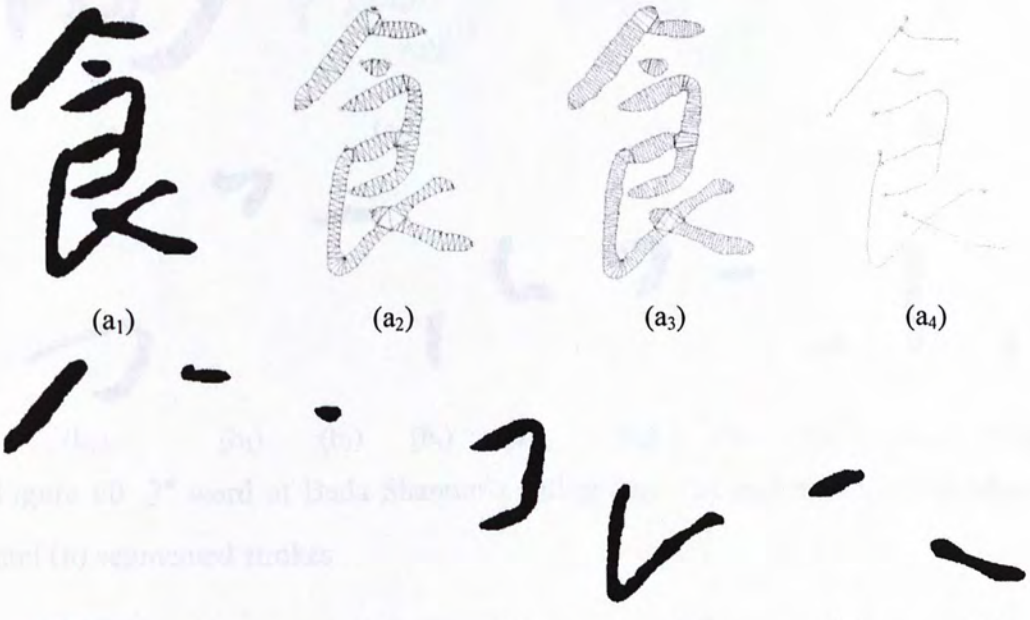
[50] "Recursive Least Square (RLS) Algorithms", Lecture Note of the Course "Identification, Estimation, and Learning" in MIT Open Courseware, http://ocw.mit.edu/OcwWeb/Mechanical-Engineering/2-160Spring-2006 /LectureNotes/index.htm

[51] K. W. Kwok, S. M. Wong, K. W. Lo, Y. Yam, "Genetic Algorithm-Based Brush Stroke Generation for Replication of Chinese Calligraphic Characte", IEEE Congress on Evolutionary Computation (CEC 2006), pp. 1057 – 1064, 2006.

[52] K. W. Kwok, K. W. Lo, S. M. Wong, Y. Yam, "Evolutionary Replication of Calligraphic Characters By A Robot Drawing Platform Using Experimentally Acquired Brush Footprint", Proceedings of IEEE International Conference on Automation Science and Engineering (CASE 2006), China, pp. 466 – 471, Oct. 2006,

APPENDIX

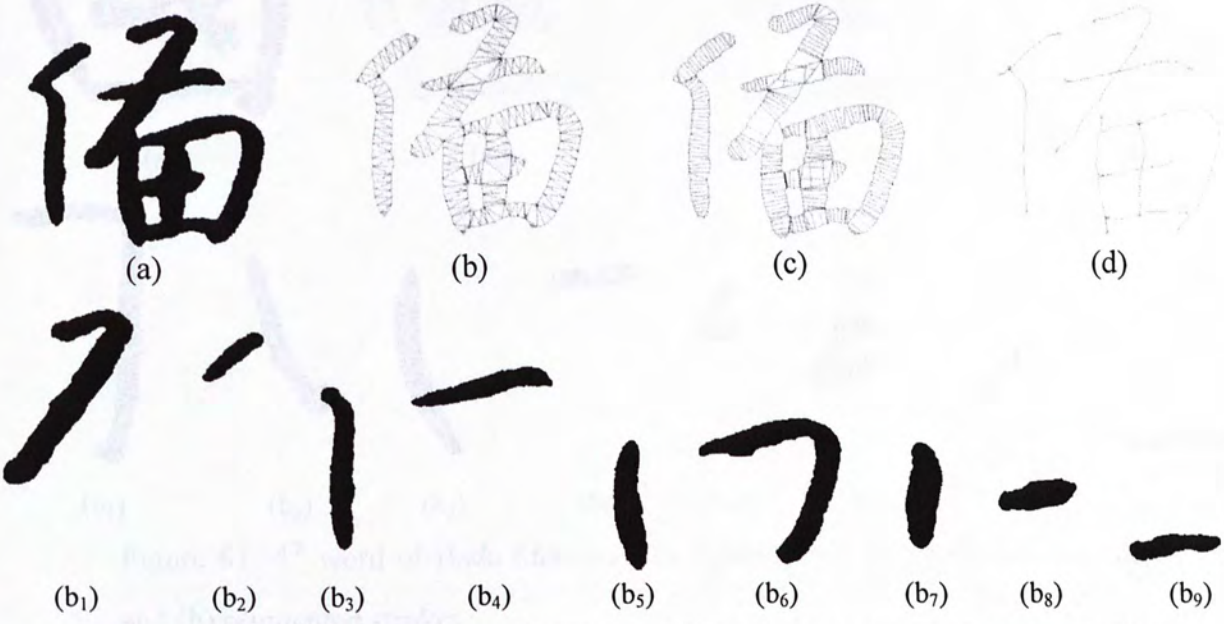
9.1. SEGMENTED STROKES OF BADA SHANREN'S CALLIGRAPHY



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7)

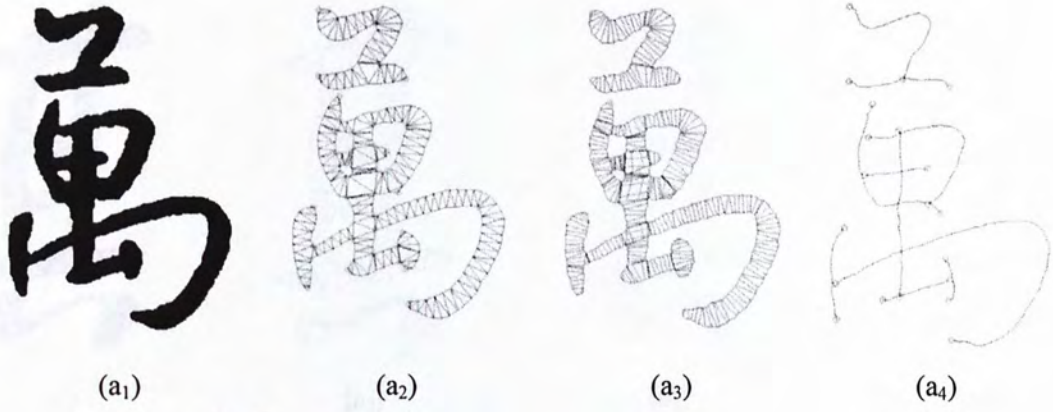
Figure 58. 1st word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



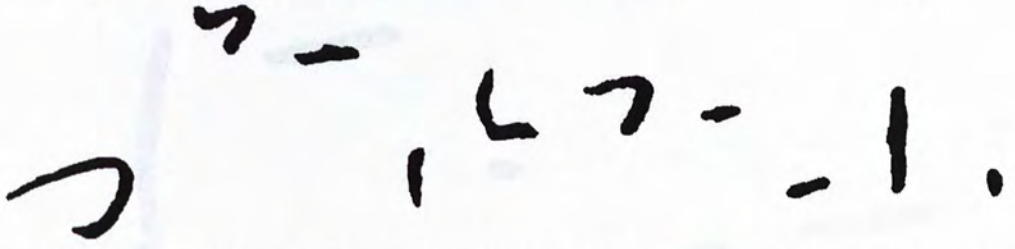
(a) (b) (c) (d)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8) (b_9)

Figure 59. 2nd word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes

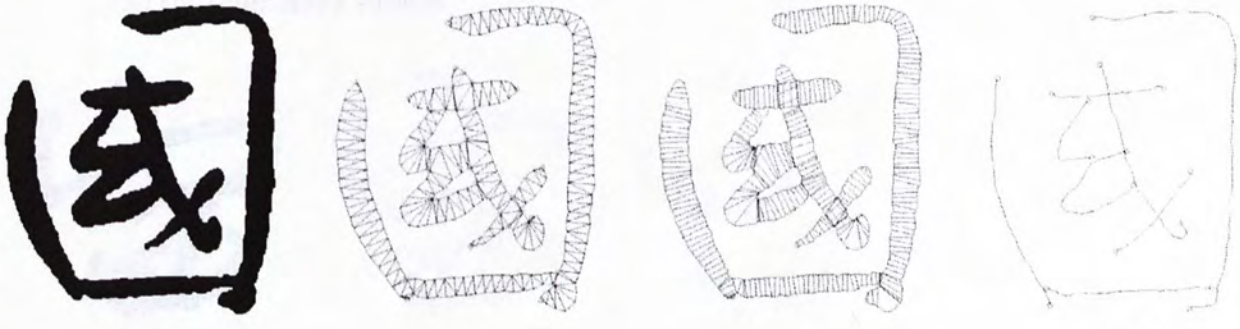


(a_1) (a_2) (a_3) (a_4)



(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8) (b_9) (b_{10})

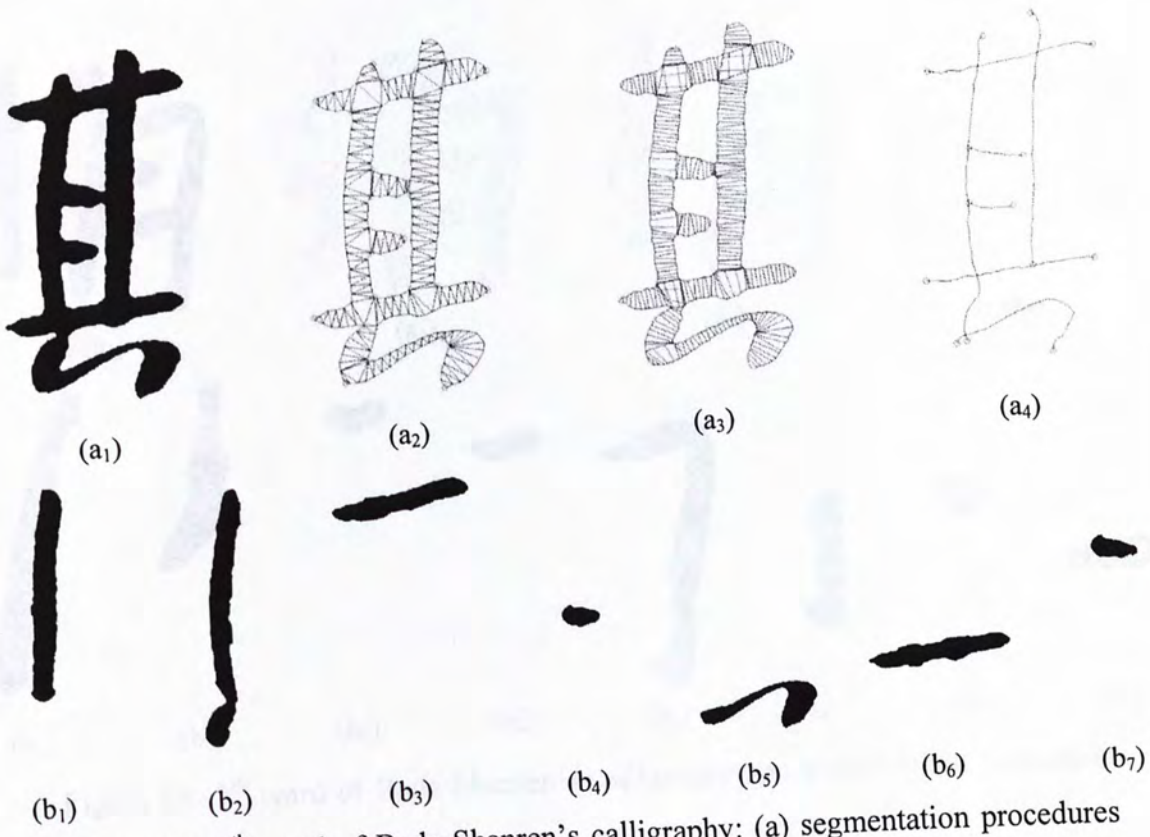
Figure 60. 3rd word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8)

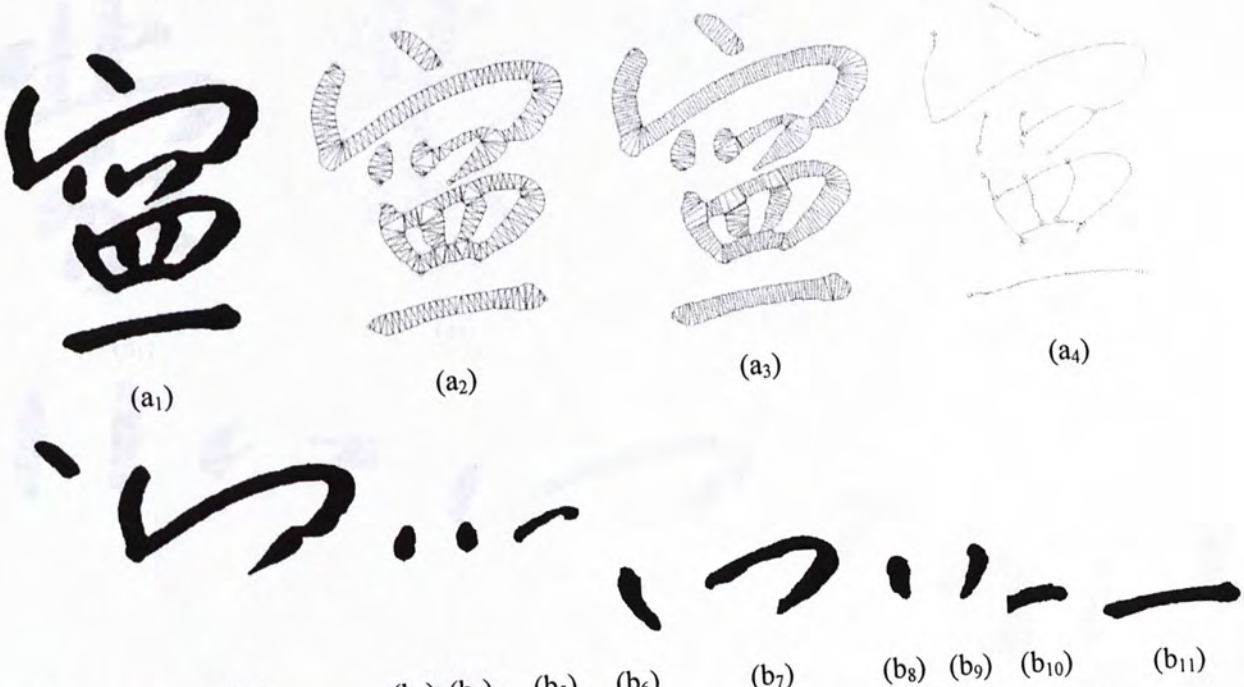
Figure 61. 4th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7)

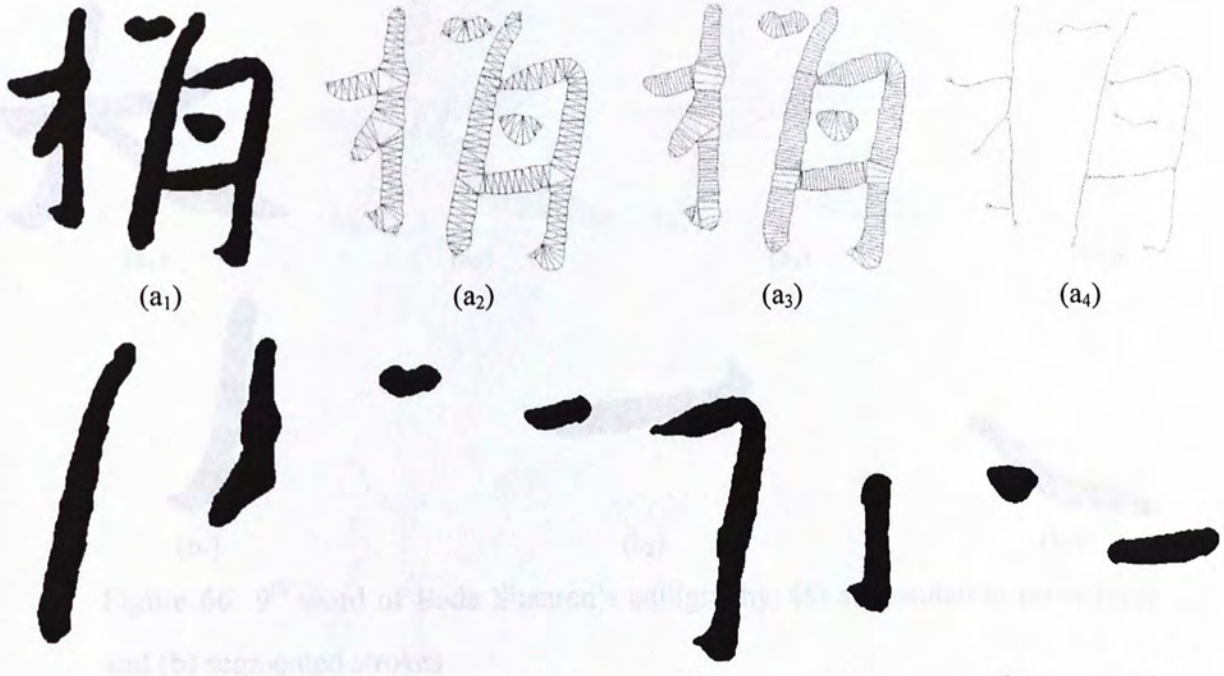
Figure 62. 5th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8) (b_9) (b_{10}) (b_{11})

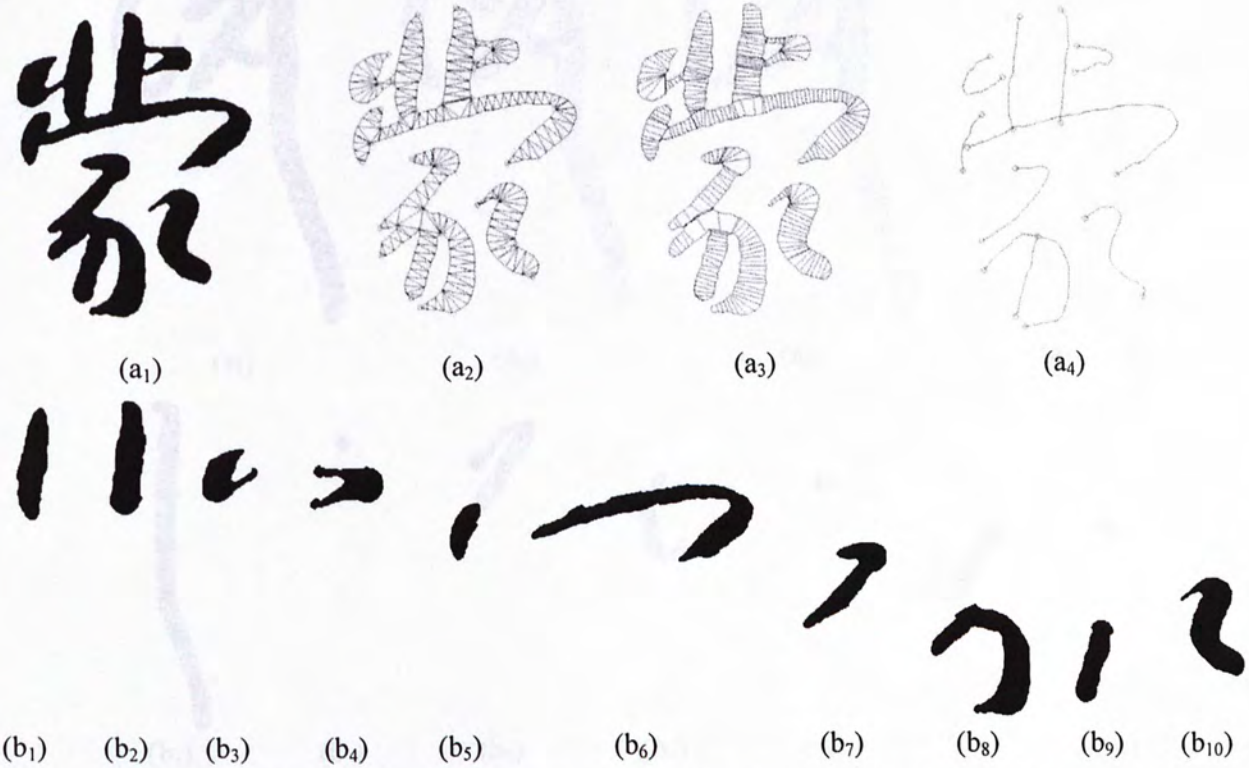
Figure 63. 6th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8)

Figure 64. 7th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8) (b_9) (b_{10})

Figure 65. 8th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes

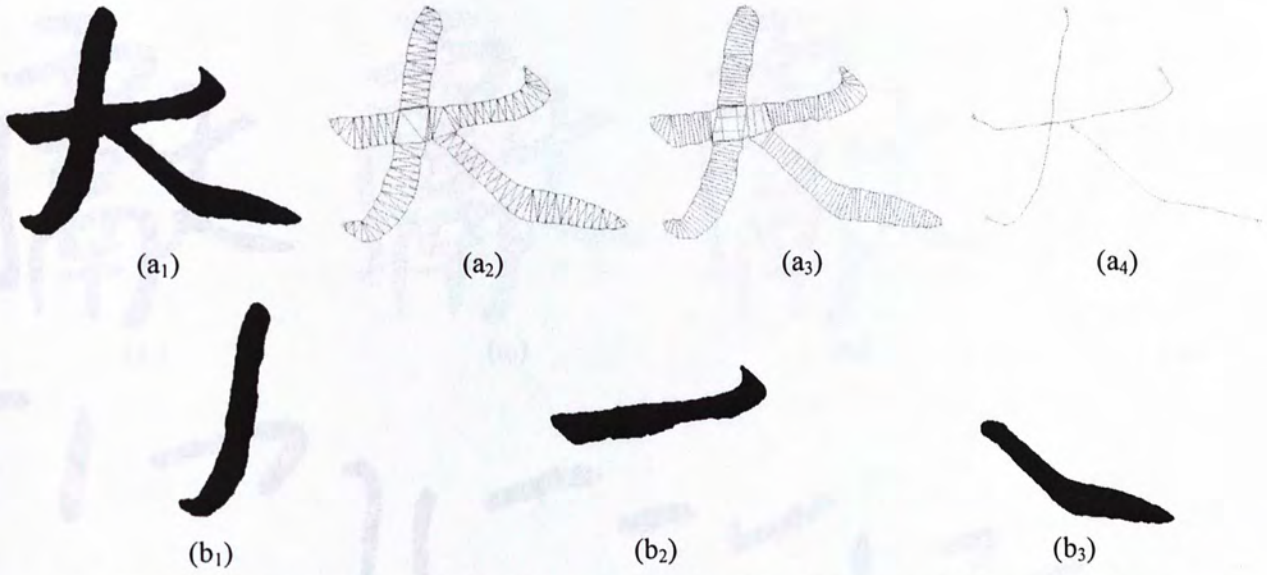


Figure 66. 9th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes

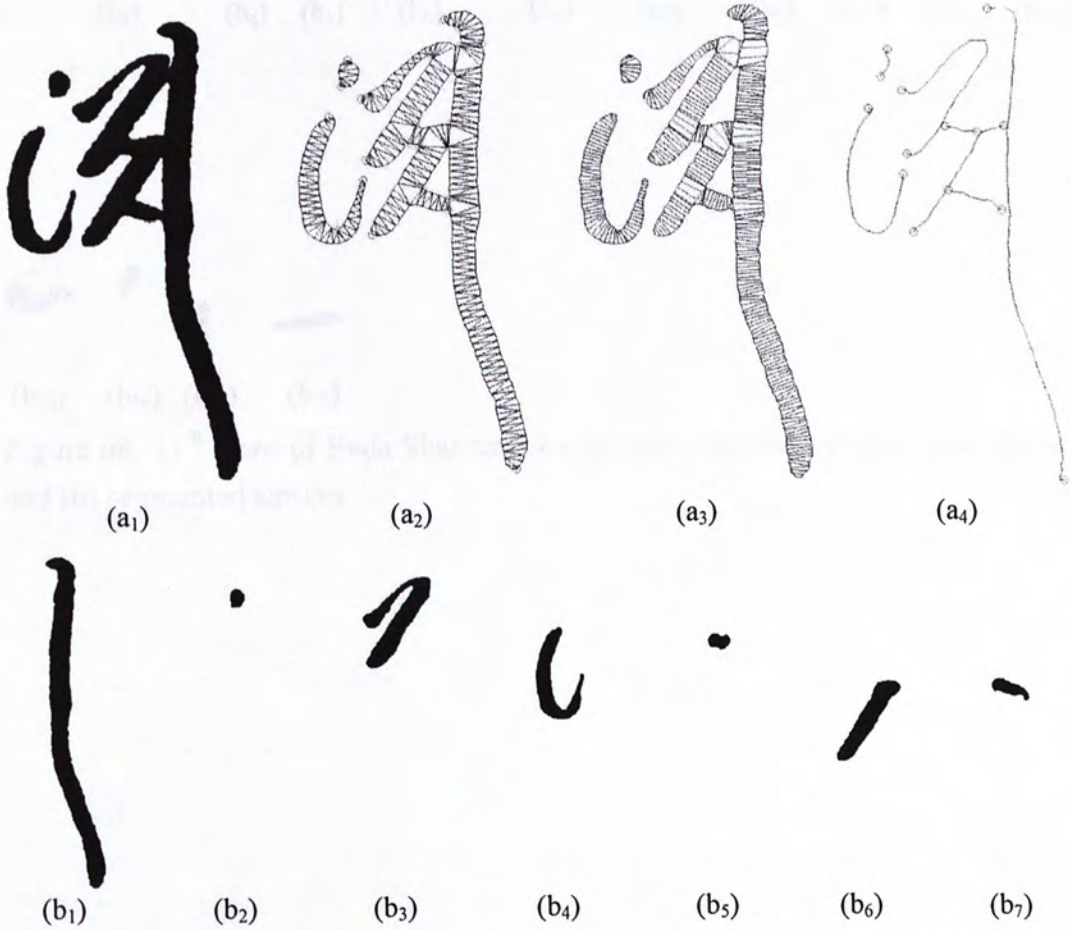
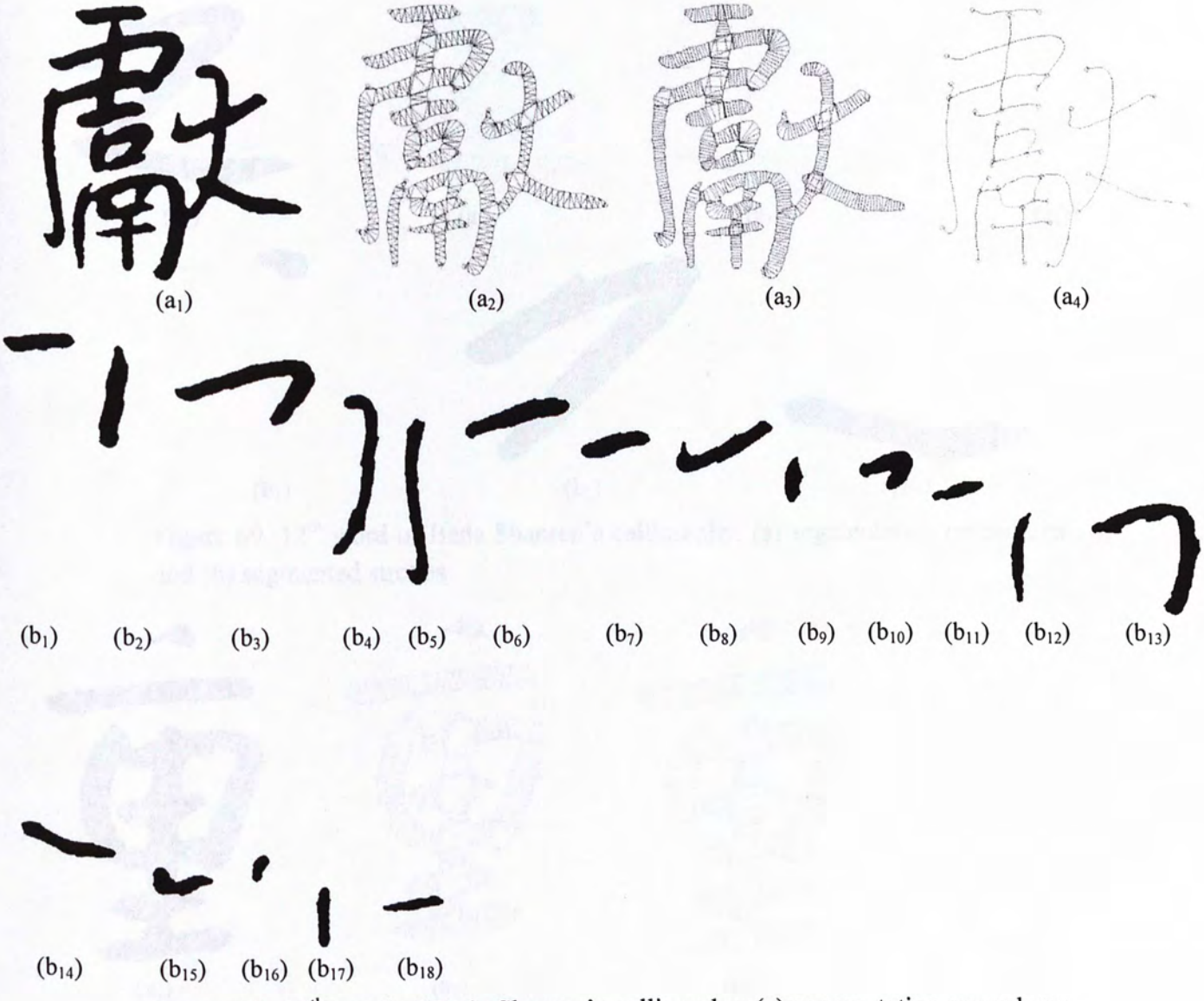


Figure 67. 10th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes

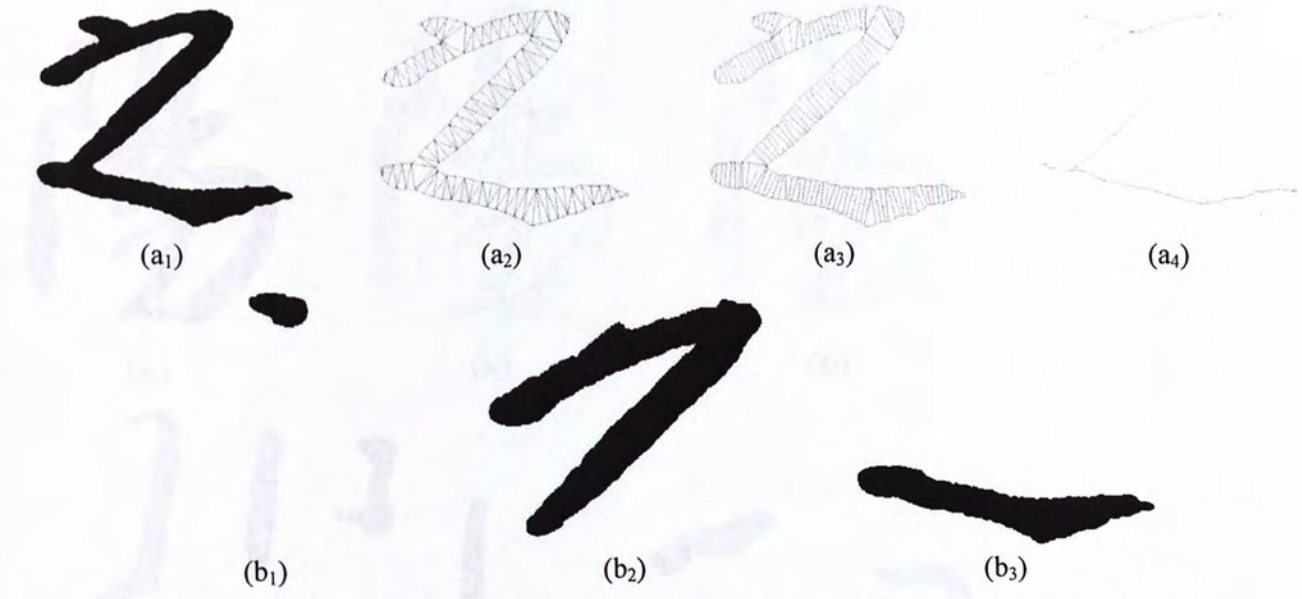


(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8) (b_9) (b_{10}) (b_{11}) (b_{12}) (b_{13})

(b_{14}) (b_{15}) (b_{16}) (b_{17}) (b_{18})

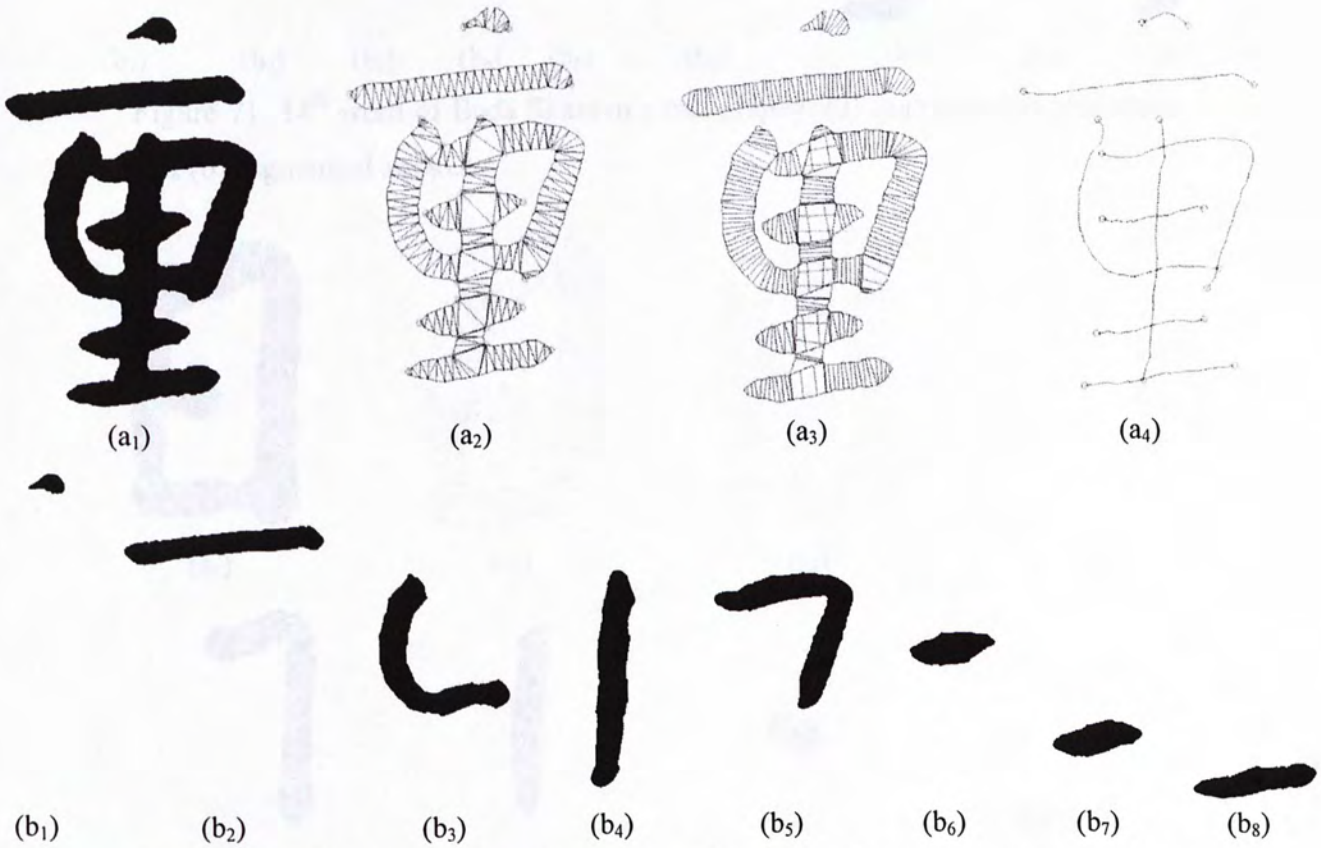
Figure 68. 11th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3)

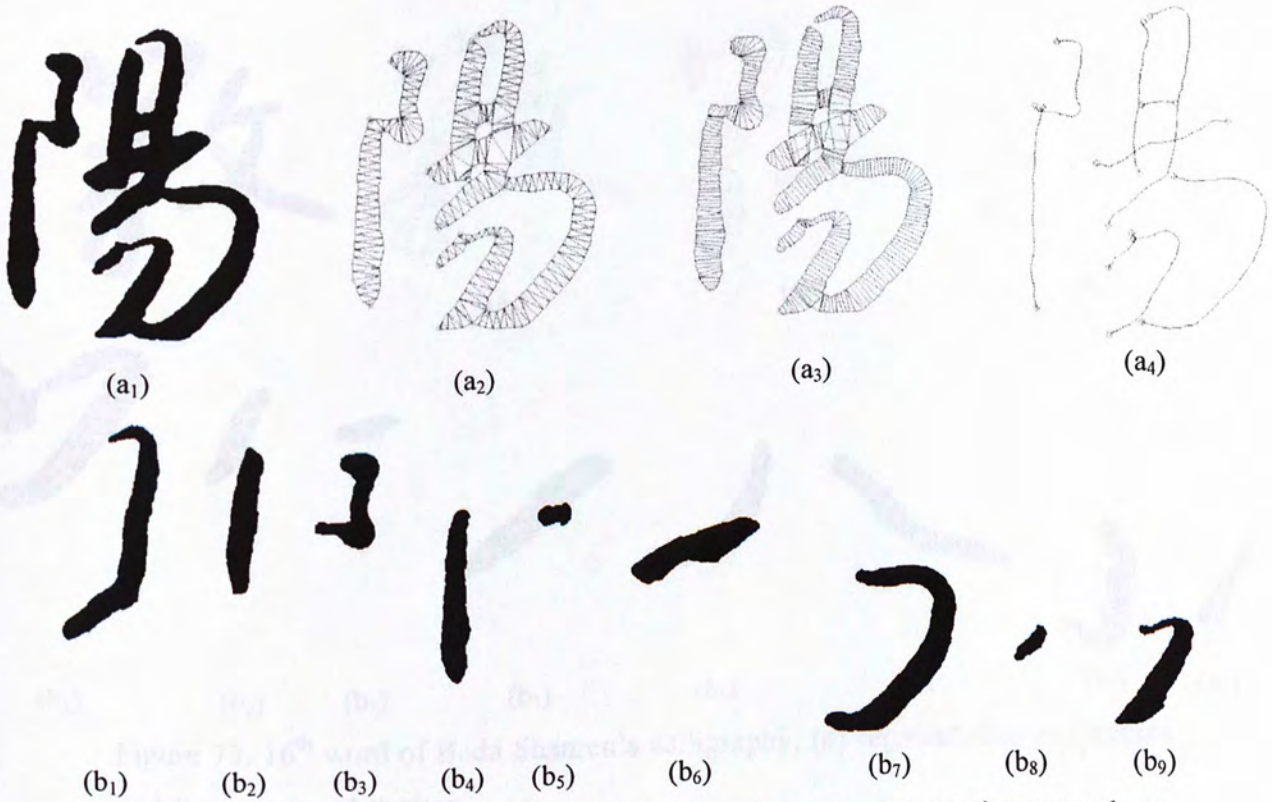
Figure 69. 12th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8)

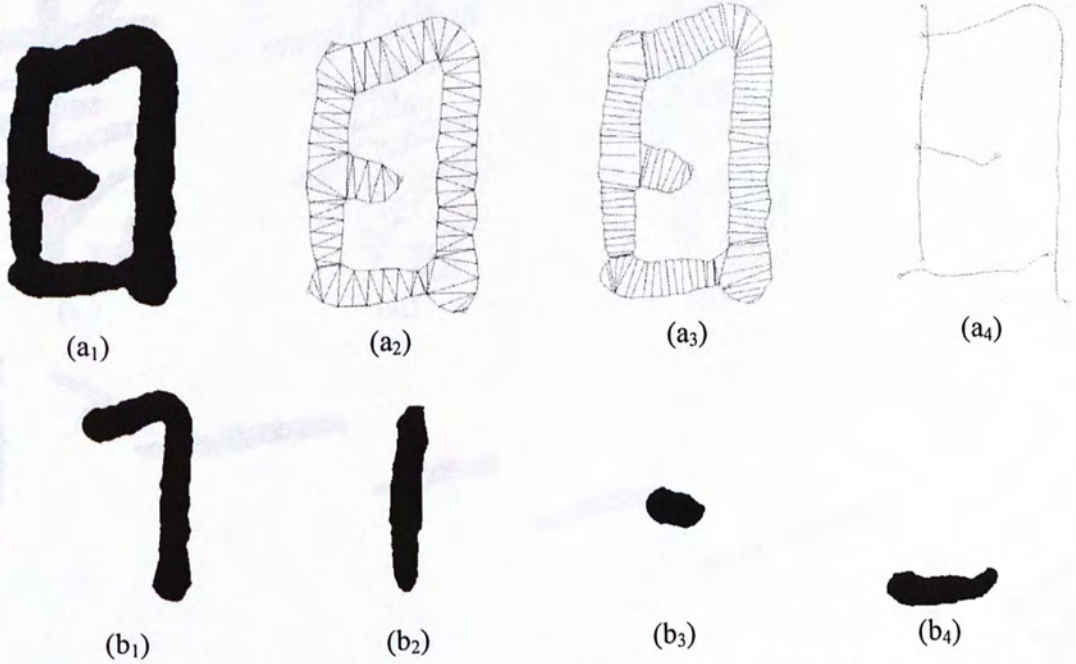
Figure 70. 13th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8) (b_9)

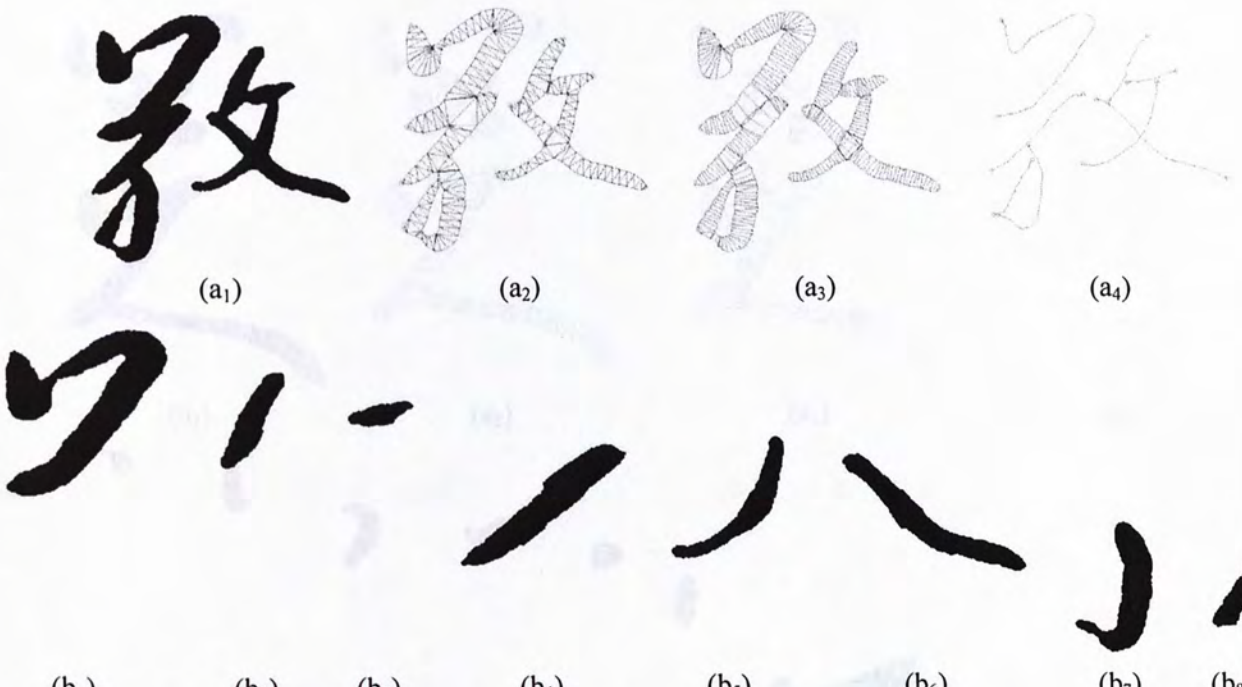
Figure 71. 14th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4)

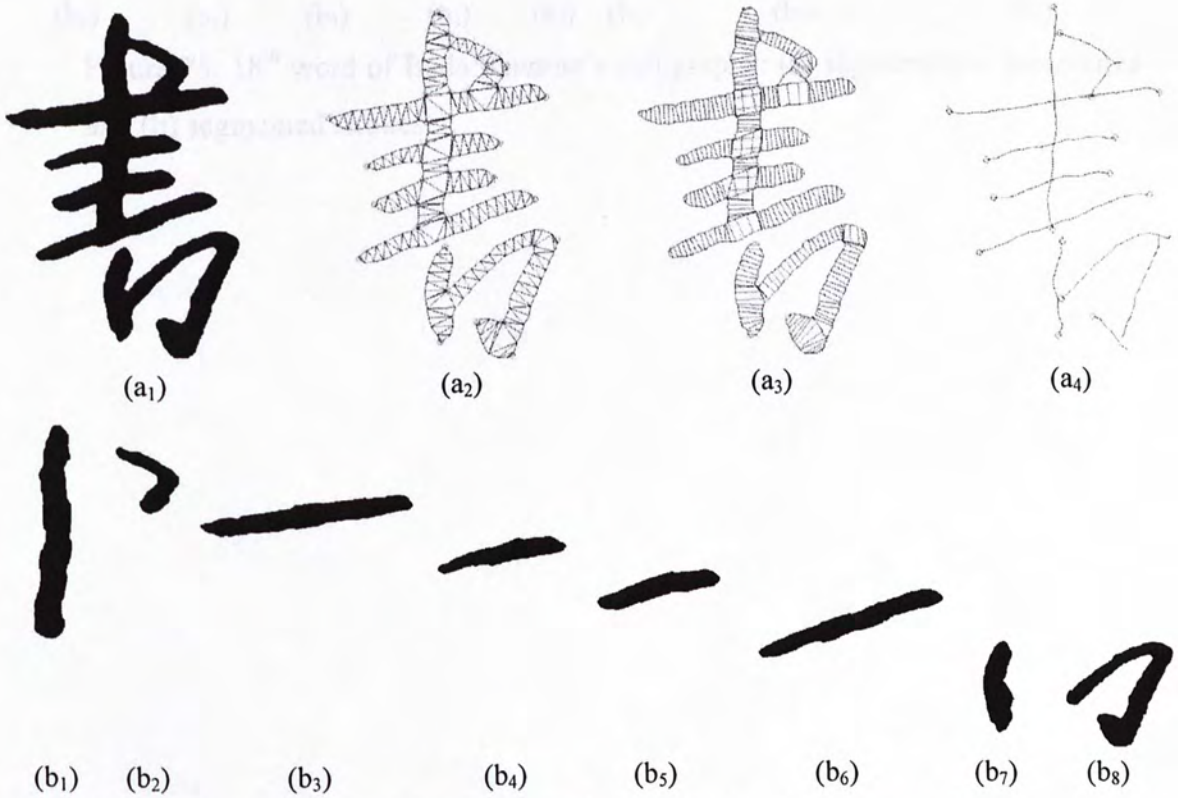
Figure 72. 15th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8)

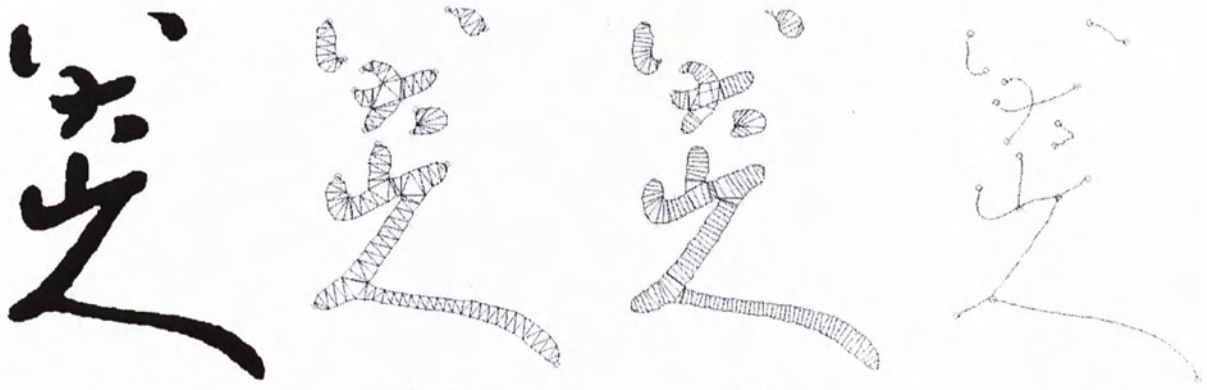
Figure 73. 16th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



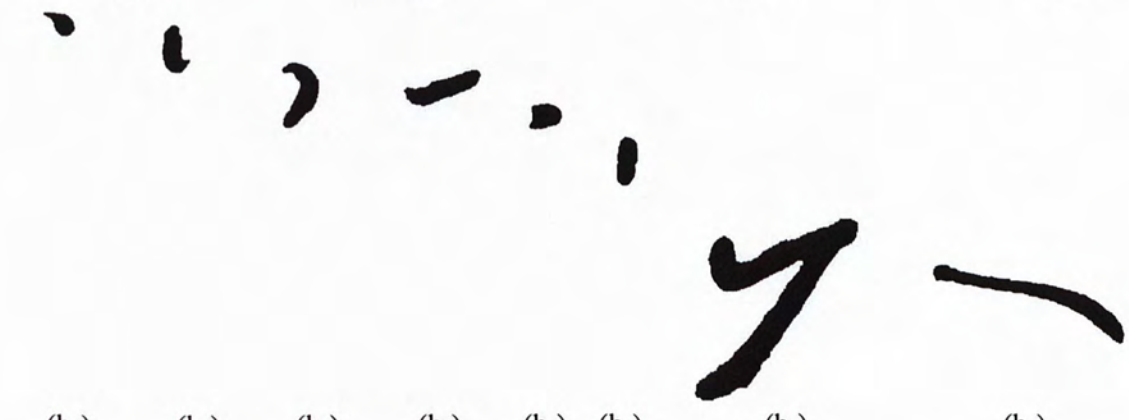
(a_1) (a_2) (a_3) (a_4)

(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8)

Figure 74. 17th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes



(a_1) (a_2) (a_3) (a_4)



(b_1) (b_2) (b_3) (b_4) (b_5) (b_6) (b_7) (b_8)

Figure 75. 18th word of Bada Shanren's calligraphy: (a) segmentation procedures and (b) segmented strokes